

**Europäisches  
Patentamt****European  
Patent Office****Office européen  
des brevets****Bescheinigung****Certificate****Attestation**

Die angehefteten Unterla-  
gen stimmen mit der  
ursprünglich eingereichten  
Fassung der auf dem näch-  
sten Blatt bezeichneten  
europäischen Patentanmel-  
dung überein.

The attached documents  
are exact copies of the  
European patent application  
described on the following  
page, as originally filed.

Les documents fixés à  
cette attestation sont  
conformes à la version  
initialement déposée de  
la demande de brevet  
européen spécifiée à la  
page suivante.

**Patentanmeldung Nr.    Patent application No.    Demande de brevet n°**

00830782.9

Der Präsident des Europäischen Patentamts;  
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets  
p.o.

**R C van Dijk**

DEN HAAG, DEN  
THE HAGUE,    26/11/01  
LA HAYE, LE

**THIS PAGE BLANK (USPTO)**



Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets

**Blatt 2 der Bescheinigung**  
**Sheet 2 of the certificate**  
**Page 2 de l'attestation**

Anmeldung Nr.:  
Application no.:  
Demande n°: 00830782.9

Anmeldetag:  
Date of filing:  
Date de dépôt: 29/11/00

Anmelder:  
Applicant(s):  
Demandeur(s):  
STMicroelectronics S.r.l.  
20041 Agrate Brianza (Milano)  
ITALY

Bezeichnung der Erfindung:  
Title of the invention:  
Titre de l'invention:

Filtering device and method for reducing noise in electrical signals, in particular acoustic signals and images

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:  
State:  
Pays:

Tag:  
Date:  
Date:

Aktenzeichen:  
File no.  
Numéro de dépôt:

Internationale Patentklassifikation:  
International Patent classification:  
Classification internationale des brevets:  
G06N3/04

Am Anmeldetag benannte Vertragsstaaten:  
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR  
Etats contractants désignés lors du dépôt:

Bemerkungen:  
Remarks:  
Remarques:

See for title page 1 of the description

this Page blank (uspto)

- 1 -

DISPOSITIVO E METODO DI FILTRAGGIO PER LA RIDUZIONE DEL  
RUMORE IN SEGNALI ELETTRICI, IN PARTICOLARE SEGNALI  
ACUSTICI E IMMAGINI

5           La presente invenzione riguarda un dispositivo ed  
un metodo di filtraggio per la riduzione del rumore in  
segnali elettrici, in particolare segnali acustici (vo-  
cali) e immagini.

          Come è noto, per rimuovere componenti indesiderate  
10 da segnali elettrici sono utilizzati filtri operanti  
secondo diverse tecniche lineari e non lineari. In par-  
ticolare, le componenti indesiderate possono essere un  
qualsiasi tipo di rumore (bianco, flicker, etc.) o al-  
tri tipi di segnali acustici o visivi sovrapposti.

15           I filtri lineari sono attualmente lo strumento più  
utilizzato per il filtraggio del rumore. I filtri FIR  
(dall'inglese Finite Impulse Response) eliminano tutte  
le armoniche di un segnale che hanno una frequenza mag-  
giore della frequenza di taglio del filtro e migliorano  
20 il rapporto segnale/rumore (SNR). Un'altra tecnica di  
filtraggio lineare è basata sulla trasformata veloce di  
Fourier (FFT-Fast Fourier Transform), in cui il segnale  
viene trasformato nel dominio delle frequenze, le armo-  
niche indesiderate sono rimosse, e quindi viene calco-  
25 lata la trasformata di Fourier inversa.

- 2 -

Per quanto riguarda le tecniche non lineari, filtri mediani sono semplici da progettare e consentono di realizzare circuiti hardware semplici. I filtri mediani si basano sul confronto dei singoli campioni di segnale in un intervallo di tempo con la media di tutti i campioni nello stesso intervallo di tempo. In base a tale confronto, i singoli campioni sono attenuati selettivamente.

Tutti questi metodi hanno lo svantaggio che, durante la rimozione del rumore, vengono rimosse anche alcune componenti del segnale originario.

Inoltre, nessuna delle attuali tecniche, sia lineare che non lineare, compreso il filtraggio mediano, è in grado di preservare fronti ripidi del segnale. Se si usa un filtro a media mobile ("moving average"), l'ampiezza della finestra deve essere molto piccola se si desidera preservare fronti ripidi. Tuttavia, se la dimensione della finestra diventa piccola, non si ha una riduzione significativa dell'energia del rumore. Se si usano filtri lineari, tutte le frequenze al di sopra della frequenza di taglio vengono eliminate, con conseguente forte distorsione del segnale.

Scopo dell'invenzione è quindi mettere a disposizione un metodo ed un dispositivo di filtraggio senza causare un sensibile deterioramento del segnale e pre-

- 3 -

servando i fronti del segnale stesso.

Secondo la presente invenzione vengono realizzati un dispositivo ed un metodo di filtraggio di segnali elettrici, come definiti nelle rivendicazioni 1 e, rispettivamente, 24.

Il filtro e il metodo descritti sono particolarmente utili nel caso di segnali aventi fronti ripidi, per i quali si desidera preservare i fronti dei segnali. Inoltre, è possibile filtrare segnali affetti da rumore bianco e non, quale ad esempio rumore flicker. Grazie al presente metodo è inoltre possibile eliminare da un segnale altri segnali sovrapposti ad esso con caratterizzazione spettrale svariata.

Il dispositivo e il metodo descritti sono basati su una rete neuro-fuzzy. Essi sono implementati con una tecnica di filtraggio a media mobile in cui i pesi per la ricostruzione finale del segnale sono calcolati in una rete neuro-fuzzy in base a specifiche regole fuzzy. Ciò consente una migliore riduzione del rumore. Le regole fuzzy operano su diverse variabili, chiamate caratteristiche ("features") di segnale. Qui sono descritte tre caratteristiche di segnale e sei regole fuzzy.

Il filtro proposto è adatto a segnali visivi o segnali acustici, anche con variazioni improvvise. Per

- 4 -

creare le regole possono essere utilizzati vari tipi di funzioni o caratteristiche di segnale. Con il metodo descritto, le caratteristiche di segnale sono correlate alla posizione del campione nella finestra di campioni considerata, alla differenza fra un campione e il campione al centro della finestra e alla differenza tra un campione e la media di campioni nella finestra. Queste caratteristiche di segnale possono influenzare notevolmente i valori dei pesi per la ricostruzione del segnale; inoltre esse possono essere calcolate in maniera relativamente semplice.

Il metodo e il filtro secondo l'invenzione comprendono inoltre un banco di filtri neuro-fuzzy. In tal modo, il segnale può essere separato in diverse sotto-bande tramite la teoria delle ondine ("wavelets"); ciascun segnale di sotto-banda può essere filtrato da una rete neuro-fuzzy e quindi le varie sotto-bande possono essere ricostruite dal banco di filtri di sintesi. Come è noto dalla teoria "wavelet", nella prima sotto-banda le caratteristiche di segnale hanno bassa frequenza, mentre nell'ultima sotto-banda le caratteristiche del segnale hanno la massima frequenza. Se deve essere cancellato del rumore non bianco (rumore colorato), questo viene approssimato da rumore bianco in ciascuna singola sotto-banda. Dato che una rete neuro-fuzzy lavora bene



- 5 -

sul rumore bianco, questa soluzione porta ad una sensibile riduzione del rumore.

La rete viene addestrata fornendo alcune configurazioni di segnali di ingresso di uscita (la configurazione del segnale di uscita che si vuole ottenere per  
5       effetto dell'evoluzione della rete è chiamata configurazione desiderata o "target"). L'algoritmo di addestramento è basato su uno dei metodi di apprendimento noti, quali il "gradient descent", un algoritmo genetico,  
10       il metodo di "annealing" simulato, la ricerca casuale o qualunque altro metodo per l'ottimizzazione di funzioni.

Per la comprensione della presente invenzione ne vengono ora descritte forme di realizzazione preferite,  
15       a puro titolo di esempio non limitativo, con riferimento ai disegni allegati, nei quali:

- la figura 1 mostra uno schema a passi mostrante l'architettura generale del filtro secondo l'invenzione;
- 20       - la figura 2 rappresenta la topologia di un passo di figura 1 relativo ad una rete neuro-fuzzy;
- le figure 3a e 3b mostrano uno schema di flusso relativo alle operazioni eseguite dal filtro di figura 1;
- 25       - la figura 4 mostra uno schema a passi di un di-

- 6 -

spositivo di filtraggio utilizzando il filtro di figura 1;

- la figura 5 mostra le funzioni di trasferimento di filtri digitali appartenenti al dispositivo di filtraggio di figura 4;

- la figura 6 mostra il diagramma a blocchi di alcuni filtri digitali appartenenti al dispositivo di filtraggio di figura 4;

- la figura 7 mostra uno schema a passi di un altro dispositivo di filtraggio utilizzando il filtro di figura 1;

- la figura 8 mostra uno schema di flusso relativo al funzionamento del dispositivo di filtro di figura 7;

- la figura 9 mostra uno schema a passi relativo all'implementazione hardware del filtro di figura 1; e

- le figure 10a, 10b, 10c mostrano rispettivamente un segnale vocale privo di rumore, lo stesso segnale sovrapposto a rumore bianco e lo stesso segnale dopo il filtraggio secondo l'invenzione.

La figura 1 mostra un filtro 1 comprendente un'unità di calcolo caratteristiche segnale 2, una rete neuro-fuzzy 3, un'unità di ricostruzione 4 e una unità di addestramento 5.

L'unità di calcolo caratteristiche segnale 2 riceve in ingresso un segnale In formato da una pluralità

- 7 -

di campioni di ingresso  $e(i)$ , memorizza, ad ogni ciclo,  $(2N+1)$  campioni di ingresso  $e(i)$  (che rappresentano una finestra di lavoro per il filtro 1) in un buffer interno, calcola le caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$  per ogni campione di ingresso  $e(i)$  sulla base di tutti i campioni di ingresso appartenenti alla finestra di lavoro (come descritto in dettaglio in seguito) e fornisce le caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$  così calcolate alla rete neuro-fuzzy.3.

La rete neuro-fuzzy 3 elabora le caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$  e genera in uscita un peso di ricostruzione  $oL3(i)$  per ogni campione di ingresso  $e(i)$ . A tale scopo, a partire dalle caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$  e per ogni campione di ingresso  $e(i)$ , la rete neuro-fuzzy 3 esegue dapprima un'operazione di fuzzificazione, quindi applica regole fuzzy prefissate e infine esegue un'operazione di defuzzificazione. Il peso di ricostruzione  $oL3(i)$  così ottenuto è quindi la somma pesata di tutti i campioni di ingresso  $e(i)$  nella stessa finestra di lavoro, come spiegato in dettaglio in seguito con riferimento alle figure 2 e 3.

L'unità di ricostruzione 4 riceve i pesi di ricostruzione  $oL3(i)$  e i campioni di ingresso  $e(i)$  e, dopo aver accumulato un sufficiente numero di campioni di

- 8 -

ingresso  $e(i)$  e di relativi pesi di ricostruzione  $oL3(i)$ , genera un campione di uscita  $u(i)$  la cui successione forma un segnale di uscita Out.

L'unità di addestramento 5 è operante solo inizialmente, in modo da addestrare la rete neuro-fuzzy 3 e modificare i pesi della rete allo scopo di ottenere un comportamento ottimale del filtro 1, come sotto descritto in dettaglio.

Le caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$ , calcolate nell'unità di calcolo caratteristiche segnale 2, sono correlate, rispettivamente, alla distanza fra ciascun campione e il campione centrale della finestra considerata, alla differenza tra un campione e il campione al centro della finestra, e alla differenza tra un campione e la media di campioni nella finestra stessa, e sono normalizzate in modo da ottenere valori compresi fra 0 e 1.

In dettaglio, data una finestra di  $(2N+1)$  campioni di ingresso  $e(i)$ , con  $i = 0, \dots, 2N$ , le caratteristiche di segnale  $X1(i)$ ,  $X2(i)$  e  $X3(i)$  per ogni campione di ingresso  $e(i)$  sono definite come:

$$X1(i) = \frac{|i - N|}{N} \quad (1)$$

$$X2(i) = \frac{|e(i) - e(N)|}{\max(diff)} \quad (2)$$

$$X3(i) = \frac{|e(i) - av|}{\max(diff\_av)} \quad (3)$$

- 9 -

in cui  $N$  è la posizione di un campione centrale  $e(N)$  nella finestra di lavoro;

$\max(diff) = \max\{e(k) - e(N)\}$  con  $k = 0, \dots, 2N$ , cioè il massimo delle differenze fra tutti i campioni di ingresso  $e(k)$  e il campione centrale  $e(N)$ ;

$av$  è il valore medio dei campioni di ingresso  $e(i)$ ;

$\max(diff\_av) = \max\{e(k) - av\}$  con  $k = 0, \dots, 2N$ , cioè il massimo delle differenze fra tutti i campioni di ingresso  $e(k)$  e il valore medio  $av$ .

La rete neuro-fuzzy 3 è una rete fuzzy a tre strati la cui rappresentazione funzionale è mostrata in figura 2, nella quale, per semplicità, l'indice  $i$  tra parentesi relativo allo specifico campione all'interno della rispettiva finestra di lavoro non è indicato. Ciò nonostante, come sopra indicato e come risulterà più chiaro dallo schema di flusso delle figure 3a, 3b, l'elaborazione neuro-fuzzy rappresentata dalla figura 2 viene ripetuta per ogni campione di ingresso  $e(i)$ .

In dettaglio, a partire dalle tre caratteristiche di segnale  $X_1$ ,  $X_2$  e  $X_3$  (o genericamente da  $l$  caratteristiche di segnale  $X_l$ ) e date  $k$  funzioni di appartenenza di tipo gaussiano per ogni caratteristica di segnale (descritte dal valore medio  $W_m(l, k)$  e dalla varianza  $W_v(l, k)$ ), viene eseguita una operazione di fuzzificazio-

-10-

ne nella quale viene valutato il livello di appartenenza delle caratteristiche di segnale  $X_1$ ,  $X_2$  e  $X_3$  rispetto a ciascuna funzione di appartenenza (qui due per ogni caratteristica di segnale, per cui  $k=2$ ; in totale, si hanno  $M = 1 \times k = 6$  funzioni di appartenenza).

In figura 2, tale operazione è rappresentata mediante sei neuroni di primo strato ("first layer neurons") 6 che, a partire dalle tre caratteristiche di segnale  $X_1$ ,  $X_2$  e  $X_3$  (genericamente indicate come  $X_l$ ) e usando come pesi il valore medio  $W_m(l,k)$  e la varianza  $W_v(l,k)$  delle funzioni di appartenenza, forniscono ciascuno un'uscita di primo strato  $oL1(l,k)$  (in seguito indicata anche come  $oL1(m)$ ) calcolata come segue:

$$oL1(l,k)=oL1(m)=\exp\left(-\left(\frac{X_l-W_m(l,k)}{W_v(l,k)}\right)^2\right) \quad (4)$$

In seguito, viene effettuata una operazione AND fuzzy utilizzando la norma del minimo, in modo da ottenere  $N$  uscite di secondo strato  $oL2(n)$ .  $N$  ad esempio è pari a 6. Come è noto, una operazione AND fuzzy utilizzando la norma del minimo si basa su regole del tipo:

if  $X_1^{(1)}$  is  $A_1^{(1)}$  and  $X_2^{(1)}$  is  $A_2^{(1)}$  and  $X_3^{(1)}$  is  $A_3^{(1)}$  then  $O$  is  $B_1^{(1)}$

...

if  $X_1^{(n)}$  is  $A_1^{(n)}$  and  $X_2^{(n)}$  is  $A_2^{(n)}$  and  $X_3^{(n)}$  is  $A_3^{(n)}$  then  $O$  is  $B_1^{(n)}$

in cui  $A_1^{(1)}$ ,  $A_2^{(1)}$ , ...,  $B_1^{(1)}$ , ecc. sono termini linguistici quali "high" "low" e il valore dell'uscita  $O$

-11-

per ogni regola è dato dal minimo dei livelli di appartenenza moltiplicato per un peso.

In pratica, con la rete neuro-fuzzy di figura 2, ogni uscita di secondo strato  $oL2(n)$  è pari al minimo fra i prodotti tra le  $M$  uscite  $oL1(m)$  dei neuroni di primo strato 6 ed un rispettivo peso di secondo strato  $W_{FA}(m, n)$ .

In figura 2, tale operazione è rappresentata mediante  $N$  neuroni di secondo strato 7 che implementano l'equazione:

$$oL2(n) = \min_m \{W_{FA}(m, n) \cdot oL1(m)\} \quad (5)$$

Infine, il terzo strato corrisponde ad un'operazione di defuzzificazione e fornisce in uscita un peso di ricostruzione  $oL3$  di tipo discreto, utilizzando  $N$  pesi di terzo strato  $W_{DF}(n)$ . Il metodo di defuzzificazione è quello del centro di gravità ed è rappresentato in figura 2 mediante un neurone di terzo strato 8 fornente il peso di ricostruzione  $oL3$  secondo l'equazione:

$$oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)} \quad (6)$$

L'unità di ricostruzione 4 attende quindi un sufficiente numero di campioni  $e(i)$  e di relativi pesi di ricostruzione  $oL3(i)$  (almeno  $2N+1$ , pari alla larghezza di una finestra di lavoro) e calcola un campione di

-12-

uscita  $u(i)$  come somma pesata dei campioni di ingresso  $e(i-j)$ , con  $j = 0 \dots 2N$ , usando i pesi di ricostruzione  $oL3(i-j)$ , secondo l'equazione:

$$u(i) = \frac{\sum_{j=0}^{2N} oL3(i-j) \cdot e(i-j)}{\sum_{j=0}^{2N} e(i-j)} \quad (7)$$

5 L'unità di addestramento 5 opera soltanto in una fase iniziale di apprendimento, durante la quale un segnale di ingresso In avente una configurazione nota viene fornito al filtro 1, il segnale di uscita Out ottenuto viene confrontato con un segnale desiderato (o  
10 di "target")  $T_g$  e la distanza fra il segnale ottenuto Out e il segnale desiderato  $T_g$  viene valutata sulla base di una funzione di conformità ("fitness"). Tale funzione di conformità può essere ad esempio il rapporto segnale/rumore quadratico avente espressione:

$$15 \quad SNR = \sum_{i=1}^T \frac{(T_g(i))^2}{(e(i) - T_g(i))^2} \quad (8)$$

in cui  $T$  rappresenta il numero totale di campioni di ingresso  $e(i)$ .

Utilizzando la funzione di conformità, e applicando un metodo per l'ottimizzazione di funzioni, quale il  
20 "gradient descent", un algoritmo genetico, il metodo di "annealing" simulato, la ricerca casuale, vengono modificati i pesi di primo strato (valore medio  $W_m(l,k)$  e varianza  $W_v(l,k)$  delle funzioni di appartenenza gaussia-



-13-

ne) e i pesi di terzo strato  $W_{DF}(n)$  e viene eseguita una nuova valutazione di conformità. I pesi di secondo strato  $W_{FA}(m,n)$  sono invece inizializzati in modo casuale e non vengono modificati. Il processo di apprendimento viene iterato fino a raggiungere un valore prefissato della funzione di conformità o fino a quando è trascorso un numero prefissato di generazioni dell'algoritmo genetico o di passi dell'algoritmo di ottimizzazione selezionato.

10        In questo modo la rete neuro-fuzzy 3 implementa un algoritmo adattativo e supera le limitazioni delle reti neurali o dei sistemi fuzzy considerati separatamente. Infatti i sistemi fuzzy non hanno capacità di apprendimento e, se la scelta delle regole fuzzy non è accurata, l'algoritmo fuzzy non si comporta in modo soddisfacente. Viceversa, utilizzando la rete neuro-fuzzy 3 ed eseguendo una fase di apprendimento preventiva, è possibile approssimare un segnale avente una struttura matematica complessa, ad esempio la voce in un ambiente rumoroso, senza conoscenza preventiva delle leggi matematiche che regolano tale sistema.

Il funzionamento del filtro 1 di figura 1 è descritto in seguito in dettaglio con riferimento alle figure 3a, 3b.

25        Inizialmente, il filtro 1 viene inizializzato,

-14-

passo 10, così come un contatore di finestre  $p$ , passo 11. Quindi il contatore di finestre  $p$  viene incrementato di  $(2N+1)$ , pari al numero di campioni di ingresso  $e(i)$  in una finestra di lavoro, passo 12 e vengono caricati i campioni di ingresso di una finestra di lavoro, compresi fra  $e(p)$  ed  $e(p+2N)$ , passo 13. Un contatore di somme campioni  $h$  viene inizializzato a 0 ed una variabile di somma campioni  $S(0)$  viene inizializzata con il valore del campione di ingresso  $e(p)$ , passo 14. Quindi, il contatore di somme campioni  $h$  viene incrementato di una unità, passo 15, e la variabile di somma campioni  $S(h)$  viene incrementata con il valore di un campione  $e(p+h)$ , passo 20. Dopo la somma di  $(2N+1)$  campioni di ingresso, uscita SI dal passo 21, la variabile di somma campioni  $S(h)$  è pari alla somma di tutti i campioni di ingresso  $e(i)$  e può essere direttamente utilizzata per il calcolo delle caratteristiche di segnale.

In seguito, utilizzando le equazioni (1), (2) e (3), e il valore della variabile di somma campioni  $S(h)$ , vengono calcolate le caratteristiche di segnale  $X_1, X_2, X_3$  per ogni campione  $e(p)-e(p+2N)$ , appartenente alla finestra di lavoro considerata, passo 22. Un contatore di campioni  $i$ , relativo ai campioni di ingresso all'interno della finestra di lavoro considerata, viene

-15-

quindi azzerato, passo 23, e successivamente incrementato di una unità, passo 24; un contatore di caratteristiche  $l$  relativo al numero di caratteristiche di segnale utilizzate (qui 3) viene azzerato, passo 25 e  
5 successivamente incrementato, passo 26; ed un contatore di funzioni  $k$  relativo al numero di funzioni di appartenenza utilizzate per ciascuna caratteristica di segnale (qui 2) viene azzerato, passo 30, e successivamente incrementato, passo 31. Quindi vengono caricati i  
10 pesi di primo strato (il valore medio  $W_m(l,k)$  e la varianza  $W_v(l,k)$ ), passo 32, e vengono calcolate le uscite di primo strato  $oL1(l,k)$ , passo 33, in base all'equazione (4).

Il caricamento dei pesi di primo strato  $W_m(l,k)$ ,  
15  $W_v(l,k)$  e il calcolo delle uscite di primo strato  $oL1(l,k)$  vengono ripetuti per tutte le funzioni di appartenenza (uscita SI dal passo 34) e per tutte le caratteristiche di segnale (uscita SI dal passo 35).

In seguito, un contatore di neuroni di secondo  
20 strato  $n$  viene azzerato, passo 39, e incrementato di una unità, passo 40; un contatore di algoritmo  $m$  viene azzerato, passo 41, e incrementato, passo 42; e vengono caricati i pesi di secondo strato  $W_{FA}(m,n)$  relativi all' $n$ -simo neurone di secondo strato 7, passi 43, 44.  
25 Quindi viene calcolata l'uscita di secondo strato

-16-

$oL2(n)$  relativa all' $n$ -simo neurone di secondo strato 7  
utilizzando l'equazione (5), passo 45, e la procedura  
dei passi 40-45 viene ripetuta per tutti gli  $n$  neuroni  
di secondo strato 7.

5 Al termine (uscita SI dal passo 50), il contatore  
di neuroni di secondo strato  $n$  viene nuovamente azzerato,  
passo 51, e incrementato di una unità, passo 52;  
vengono caricati in successione tutti i pesi di terzo  
strato  $W_{DF}(n)$ , passi 53, 54; al termine del caricamento  
10 (uscita SI dal passo 54), viene calcolato il peso di  
ricostruzione  $oL3(i)$  secondo l'equazione (6), passo 55.

I pesi di ricostruzione  $oL3(i)$  così calcolati e i  
rispettivi campioni di ingresso  $e(i)$  vengono memorizzati,  
passo 60, e la procedura descritta dei passi 24-60  
15 viene ripetuta per successivi campioni di ingresso  
 $e(i)$ , fino a raggiungere  $(2N+1)$  campioni di ingresso  
 $e(i)$  (uscita SI dal passo 61a).

In seguito viene azzerato un contatore di somma  $j$ ,  
passo 62, e il campione di ingresso  $e(i)$  viene sommato  
20 ai precedenti  $2N$  campioni di ingresso  $e(i-j)$ ; la somma  
è memorizzata in una variabile  $E$ , passo 63.

Dopo aver eseguito la somma di  $(2N+1)$  campioni di  
ingresso, viene calcolato un campione di uscita  $u(i)$ ,  
secondo l'equazione (7), passo 65.

25 L'intero ciclo descritto dai passi 12-65 viene ri-

BEST AVAILABLE COPY

-17-

petuto fintanto che sono presenti campioni di ingresso e(i); al termine (uscita SI dal passo 70) termina la procedura di elaborazione dei campioni.

Secondo un altro aspetto dell'invenzione, il fil-  
5 traggio è basato su una analisi a multirisoluzione ottenuta mediante un banco di filtri in quadratura. La teoria delle ondine ("wavelets") fornisce la base teorica della multirisoluzione.

Come è noto, una analisi a multirisoluzione defi-  
10 nisce un gruppo di sottospazi annidati di uno spazio funzionale sommabile in modo quadratico ("square summable function space", cioè lo spazio delle funzioni di energia finita), ampiamente noto nella fisica e nell'ingegneria elettrica. In base a tale analisi, una  
15 proiezione di una funzione in uno di questi sotto spazi isola i dettagli "più grossolani" della funzione, mentre la proiezione della funzione nel complemento ortogonale di questo sottospazio isola i dettagli "più fini" della funzione. La procedura può essere iterata in  
20 modo da ottenere una piramide. Dalla teoria "wavelets" è noto che la stessa operazione può essere eseguita tramite un banco di filtri FIR, in cui ciascun filtro FIR è seguito da un riduttore di campioni ("subsampler"). Il segnale, così suddiviso in sottobande, può  
25 essere ricostruito tramite un banco di filtri comple-

## BEST AVAILABLE COPY

-18-

mentari, dotati ciascuno di un incrementatore di campioni ("upsampler").

Una prima forma di implementazione di tale soluzione è mostrata in figura 4, ed è relativa ad un filtro multirisoluzione 80 generante due sottospazi e utilizzando una prima coppia di filtri FIR QMF (dall'inglese Finite Impulse Response Quadrature Mirror Filter) per l'analisi ed una seconda coppia di filtri FIR QMF per la sintesi.

10 In dettaglio, il filtro multirisoluzione 80 riceve, su un ingresso 81, un segnale di ingresso  $s_1(t)$ . Il segnale di ingresso  $s_1(t)$  è fornito a due filtri FIR di ingresso  $H_0$ ,  $H_1$  che eseguono una operazione di convoluzione in modo da fornire in uscita un rispettivo segnale filtrato  $\tilde{e}_0$ ,  $\tilde{e}_1$  pari a:

$$\tilde{e}_0(t) = \sum_{r=0} h_0(r) \cdot i(t-r) \quad (9)$$

$$\tilde{e}_1(t) = \sum_{r=0} h_1(r) \cdot i(t-r) \quad (10)$$

in cui  $M$  è l'ordine dei filtri  $H_0$ ,  $H_1$ ;  $\tilde{e}_0(t)$ ,  $\tilde{e}_1(t)$  è il  $t$ -esimo campione della rispettiva sequenza di uscita;  $s_1(t)$  è il  $t$ -esimo campione della sequenza di ingresso;  $h_0(r)$ ,  $h_1(r)$  è il tappo ("tap")  $t$ -esimo del filtro FIR di ingresso  $H_0$ ,  $H_1$ , in cui

$$h_1(r) = (-1)^{r-1} h_0(2I-r+1)$$

con  $I$  numero intero.

25 I filtri FIR di ingresso  $H_0$ ,  $H_1$  hanno funzioni di

-19-

trasferimento mostrate in figura 5 e definiscono quindi  
rispettivamente un filtro passa-basso ed un filtro pas-  
sa-alto ed hanno struttura mostrata in figura 6, in cui  
il generico tappo  $h(r)$  corrisponde a  $h_0(r)$  o a  $h_1(r)$ , a  
5 seconda che si tratti del filtro FIR di ingresso  $H_0$  o  
del filtro FIR di ingresso  $H_1$ .

Le uscite dei filtri FIR di ingresso  $H_0$ ,  $H_1$  sono  
collegate ciascuna ad una rispettiva unità di riduzione  
campioni ("subsampler") 84, 85 che scarta i campioni  
10 dispari dal segnale di ingresso  $\tilde{e}_0(t)$ ,  $\tilde{e}_1(t)$  e mantiene  
solo i campioni pari, generando un rispettivo segnale  
 $e_0(t)$ ,  $e_1(t)$ . Le uscite delle unità di riduzione campio-  
ni 84, 85 sono collegate ciascuna ad un rispettivo fil-  
tro neuro-fuzzy 86, 87. I filtri neuro-fuzzy 86, 87  
15 hanno entrambi la struttura mostrata in figura 1.

I segnali di uscita  $u_0(t)$ ,  $u_1(t)$  dei filtri neuro-  
fuzzy 86, 87 sono forniti ciascuno ad una rispettiva  
unità di incremento campioni ("upsampler") 88, 89 che  
genera una rispettiva sequenza di uscita  $u_{0u}(t)$ ,  $u_{1u}(t)$   
20 inserendo un campione zero fra ciascuna coppia di cam-  
pioni del rispettivo segnale di uscita  $u_0(t)$ ,  $u_1(t)$  dei  
filtri neuro-fuzzy 86, 87. Le uscite delle unità di in-  
cremento campioni 88, 89 sono collegate ciascuna ad un  
rispettivo filtro FIR di uscita  $G_0$ ,  $G_1$ , aventi anch'essi  
25 una rispettiva funzione di trasferimento, date dalla

BEST AVAILABLE COPY

-20-

(9) e dalla (10).

Infine, i segnali di uscita dei filtri FIR di uscita  $G_0$ ,  $G_1$  sono sommati fra loro, campione per campione, da un sommatore 92.

5        La figura 7 mostra un filtro multirisoluzione 95  
utilizzante otto sottospazi. In dettaglio, il segnale di ingresso  $s_1(t)$  viene inizialmente fornito a due primi filtri FIR di sintesi  $H_{001}$  e  $H_{102}$ , rispettivamente di tipo passa-basso e passa-alto, e quindi viene ridotto di  
10    campioni ("downsampled") in due prime unità di riduzione campioni 96, 97, analogamente a quanto descritto per le unità 85, 86 di figura 4. Le sequenze di campioni così ottenute sono fornite ciascuna a due filtri di sintesi (e quindi complessivamente a quattro secondi  
15    filtri FIR di sintesi  $H_{011}$ ,  $H_{112}$ ,  $H_{013}$  e  $H_{114}$ ). Le uscite dei secondi filtri FIR di sintesi  $H_{011}$ ,  $H_{111}$ ,  $H_{013}$  e  $H_{114}$  sono quindi fornite a quattro seconde unità di riduzione campioni 100-103 e ciascuna sequenza così ottenuta viene fornita a due terzi filtri FIR di sintesi (e  
20    quindi complessivamente a quattro terzi filtri FIR di sintesi  $H_{021}$ ,  $H_{122}$ ,  $H_{023}$ ,  $H_{124}$ ,  $H_{025}$ ,  $H_{126}$ ,  $H_{027}$  e  $H_{128}$ ), generando otto sequenze di campioni. Le otto sequenze di campioni sono quindi fornite a otto terze unità di riduzione campioni 107-114 ed elaborate in rispettivi  
25    filtri neuro-fuzzy 120-127 aventi la struttura di figu-



-21-

ra 1. Le sequenze di campioni presenti sulle uscite dei filtri neuro-fuzzy 120-127 sono quindi incrementate attraverso unità di incremento campioni 130-137 e fornite a rispettivi primi filtri FIR di sintesi  $G_{021}$ ,  $G_{122}$ ,  $G_{023}$ ,  
5  $G_{124}$ ,  $G_{025}$ ,  $G_{126}$ ,  $G_{027}$  e  $G_{128}$ . Le sequenze di campioni così ottenute sono sommate a due a due tramite quattro sommatore 140-143 (con un processo inverso a quello subito a valle delle seconde unità di riduzione dei campioni 100-103), fornite a quattro unità di incremento campio-  
10 ni 146-149 e filtrate nuovamente attraverso quattro secondi filtri FIR di sintesi  $G_{011}$ ,  $G_{112}$ ,  $G_{013}$  e  $G_{114}$ .

Le sequenze di campioni così ottenute sono sommate a due a due tramite due sommatore 150, 151 (con un processo inverso a quello subito a valle delle prime unità  
15 di riduzione dei campioni 96, 97), incrementate tramite due unità di incremento campioni 154, 155; filtrate attraverso due terzi filtri FIR di sintesi  $G_{001}$  e  $G_{102}$  e infine sommate in un sommatore 160 in modo da fornire il segnale di uscita  $s_o(t)$ .

20 La figura 8 mostra uno schema di flusso relativo alla sequenza di passi eseguiti con un filtro multirisoluzione con un numero arbitrario, prefissato, di suddivisioni in sottospazi.

Inizialmente, passo 160, vengono caricati i cam-  
25 pioni del segnale di ingresso  $s_i(t)$ ; quindi un contatore

-22-

di divisione spazi  $j$  viene inizializzato a  $-1$ , passo 161, e incrementato di una unità, passo 162; un contatore di sottospazi  $k$  viene inizializzato a zero, passo 163, e incrementato di una unità, passo 164. Quindi, i  
5 campioni del segnale di ingresso  $s_i(t)$  vengono filtrati utilizzando il filtro  $H_{0jk}$  (quindi, alla prima iterazione, utilizzando il filtro  $H_{001}$ ), passo 165, e i campioni filtrati vengono ridotti di numero ("downsampled"), passo 166. Quindi il contatore di sottospazi  $k$  viene  
10 incrementato, passo 170; i campioni del segnale di ingresso  $s_i(t)$  vengono filtrati utilizzando il filtro  $H_{1jk}$  (quindi, alla prima iterazione, utilizzando il filtro  $H_{102}$ ), passo 171, e i campioni filtrati vengono ridotti di numero, passo 172.

15 Viene verificato se il contatore di sottospazi  $k$  è pari a  $2^{(j+1)}$ , in caso negativo viene ripetuto il ciclo comprendente i passi 164-172 (nelle iterazioni successive alla prima, il filtraggio viene eseguito sui campioni ottenuti nella iterazione precedente con il fil-  
20 tro  $H_{0(j-1)(k-2)}$  o  $H_{1(j-1)(k-2)}$ ), in caso positivo (uscita SI dal passo 173) si verifica se la suddivisione in sottospazi è terminata (cioè se il contatore di divisione spazi  $j$  ha raggiunto il massimo previsto), passo 174. In caso negativo, si ritorna al passo 162; in caso po-  
25 sitivo, tutte le sequenze di campioni ottenute vengono

-23-

filtrate utilizzando il filtro neuro-fuzzy di figura 1, passo 175.

Quindi il contatore di divisione spazi  $j$  viene inizializzato al suo valore massimo più una unità, passo 180, e quindi decrementato di una unità, passo 181; il contatore di sottospazi  $k$  viene inizializzato a zero, passo 182, e incrementato di una unità, passo 183. Quindi, la prima sequenza di campioni in uscita dal primo filtro neuro-fuzzy (120 in figura 7) viene incrementata ("upsampled"), passo 184, e filtrata utilizzando un filtro  $G_{0,j,k}$  (quindi, alla prima iterazione, usando il filtro  $G_{0,21}$ ), passo 185. Quindi il contatore di sottospazi  $k$  viene nuovamente incrementato, passo 190, una seconda sequenza di campioni in uscita da un secondo filtro neuro-fuzzy (121 in figura 7) viene incrementata, passo 191, e filtrata utilizzando un filtro  $G_{1,j,k}$  (quindi, alla prima iterazione, usando il filtro  $G_{1,21}$ ), passo 192. I campioni in uscita dai filtri  $G_{0,j,k}$  e  $G_{1,j,k}$  vengono poi sommati, passo 193.

Si verifica quindi se il contatore di sottospazi  $k$  è pari a  $2^{(j-1)}$ , passo 194; in caso negativo viene ripetuto il ciclo comprendente i passi 183-193 (elaborando le sequenze di campioni in uscita dai successivi filtri neuro-fuzzy), in caso positivo (uscita SI dal passo 194), si verifica se si è arrivati alla fine, passo

-24-

195; in caso negativo si ritorna al passo 181, decrementando il contatore di divisione spazi  $j$  ed elaborando le sequenze di campioni precedentemente incrementate, filtrate e sommate. L'anello definito dai passi  
5 181-194 viene ripetuto fino ad ottenere una unica sequenza di campioni, corrispondente al segnale di uscita  $s_o(t)$ , uscita SI dal passo 195.

La figura 9 mostra l'implementazione hardware del filtro neuro-fuzzy 1 di figura 1. In dettaglio, il filtro neuro-fuzzy 1 comprende una memoria dati 200, tre  
10 memorie di lavoro 201-203, un modulo per il calcolo delle caratteristiche di segnale 205, una unità di comando 206, una unità di memoria delle uscite di primo strato 207, una unità di calcolo delle uscite di secondo  
15 strato 208, una unità di calcolo dei pesi di ricostruzione 209 e un'unità di ricostruzione 210.

La memoria dati 200 memorizza i  $(2N+1)$  campioni  $e(i)$  di ciascuna finestra di lavoro ed è costituita da  $(2N+1)$  registri, ciascuno a 16 bit.

20 Le memorie di lavoro 201-203 sono memorie non volatili, ad esempio a ROM, PROM, EPROM, EEPROM o flash.

In particolare, la prima memoria di lavoro 201 memorizza la prima caratteristica di segnale  $X_1(i)$  e comprende  $(2N+1)$  locazioni di memoria a 16 bit. Dato che  
25 il valore della prima caratteristica di segnale  $X_1(i)$

-25-

per il campione i-esimo è costante nelle diverse finestre di campioni, come risulta evidente dalla (1), il contenuto della prima memoria di lavoro 201 non deve essere modificato durante la fase di apprendimento o  
5 durante il funzionamento del filtro neuro-fuzzy 1 e la prima memoria di lavoro 201 può essere realizzata in una qualsiasi delle tecnologie sopra indicate.

La seconda memoria di lavoro 202 memorizza il valore delle due curve gaussiane descritte dalla (2) in  
10 funzione del valore delle caratteristiche di segnale  $X_1$ ,  $X_2$  e  $X_3$ . Dato che il valore di tale curve gaussiane dipende dai pesi di secondo strato  $W_m(1,k)$ ,  $W_v(1,k)$ , nel caso sia prevista una fase di apprendimento, la seconda memoria di lavoro 202 deve essere di tipo programmabile,  
15 le, ad esempio di tipo EPROM, EEPROM o flash. Per evitare l'uso di una memoria di dimensioni eccessive, le funzioni gaussiane (che rappresentano le funzioni di appartenenza delle caratteristiche di segnale  $X_1$ ,  $X_2$ ,  $X_3$ , come sopra discusso) sono memorizzate come valori  
20 discreti, in base al livello di accuratezza desiderato. Ad esempio, nel caso che le funzioni di appartenenza assumano 256 valori, con una accuratezza di 16 bit per ciascun valore, considerando due gruppi di regole fuzzy per ogni caratteristica di segnale  $X_1$ ,  $X_2$ ,  $X_3$ , la se-  
25 conda memoria di lavoro 202 deve presentare una capaci-

-26-

tà di memoria pari a  $256 \times 16 \times 6$  bit. La seconda memoria di lavoro 202 viene quindi indirizzata a partire dal valore corrente (relativo al campione  $i$ -esimo) delle caratteristiche di segnale  $X_1$ ,  $X_2$ ,  $X_3$ , fornite dalla  
5 prima memoria di lavoro 201 e dal modulo per il calcolo delle caratteristiche 205, e fornisce in uscita (all'unità di memoria delle uscite di primo strato 207) il valore delle sei uscite di primo strato  $oLl(m)$ .

La terza memoria di lavoro 203 memorizza i pesi di  
10 secondo strato  $W_{FA}(m,n)$  e i pesi di terzo strato  $W_{DF}(n)$ . Dato che i pesi di terzo strato  $W_{DF}(n)$  vengono generalmente modificati durante la fase di apprendimento, la terza memoria di lavoro 203 è di tipo programmabile, analogamente alla seconda memoria di lavoro 202. In  
15 dettaglio, se si hanno  $M \times N$  pesi di secondo strato  $W_{FA}(m,n)$  e  $N$  pesi di terzo strato  $W_{DF}(n)$ , la memoria di lavoro 203 comprende  $(16 \times M \times N + 16 \times N)$  bit.

Il modulo per il calcolo delle caratteristiche 205 comprende una rete hardware non mostrata, rappresentata  
20 divisa in una prima parte 205a di calcolo della seconda caratteristica di segnale  $X_2(i)$  e in una seconda parte 205b di calcolo della terza caratteristica di segnale  $X_2(i)$ . Nel modulo per il calcolo delle caratteristiche 205 vengono eseguite le operazioni descritte dalle  
25 equazioni (2) e (3) e comprende una memoria di campioni

-27-

(avente  $2N+1$  locazioni), un'unità per il calcolo del valore medio  $av$ , un'unità per il calcolo del massimo tra le differenze fra tutti i campioni di ingresso e il campione centrale  $\max(diff)$ , un'unità per il calcolo  
5 del massimo tra le differenze fra tutti i campioni di ingresso e il valore medio  $\max(diff\_av)$  e un'unità per il calcolo delle frazioni definite dalle equazioni (2) e (3).

L'unità di memoria delle uscite di primo strato  
10 207 è formata da sei registri 212 che memorizzano le uscite di primo strato  $oL1(m)$  fornite dalla seconda memoria di lavoro 202.

L'unità di calcolo delle uscite di secondo strato  
208 è formata da due moduli, e precisamente un primo  
15 modulo moltiplicativo 208a ed un modulo di minimo 208b.  
In dettaglio, il primo modulo moltiplicativo 208a è formato da sei prime unità moltiplicative 213 che moltiplicano, ciascuna, una rispettiva uscita di primo strato  $oL1(m)$  (fornita dall'unità di memoria delle  
20 uscite di primo strato 207) con  $n$  rispettivi pesi di secondo strato  $W_{FA}(m,n)$  (forniti dalla seconda memoria di lavoro 203); il secondo modulo moltiplicativo 208b è formato da sei unità di minimo 214 che, a partire dai sei rispettivi prodotti  $oL1(m) \times W_{FA}(m,n)$ , ne calcolano il  
25 minimo, fornendo in uscita una rispettiva uscita di se-

-28-

condo strato  $oL2(n)$ .

L'unità di calcolo dei pesi di ricostruzione 209 è formata da due moduli, e precisamente un secondo modulo moltiplicativo 209a ed un modulo di defuzzificazione 5 209b. In dettaglio, il secondo modulo moltiplicativo 209a è formato da sei seconde unità moltiplicative 215 che moltiplicano una rispettiva uscita di secondo strato  $oL2(n)$ , fornita dall'unità di calcolo delle uscite di secondo strato 208, per un rispettivo peso di terzo 10 strato  $W_{DF}(n)$ , fornito dalla terza memoria di lavoro 203. Il modulo di defuzzificazione 209b calcola i pesi di ricostruzione  $oL3$ , sommando i prodotti forniti dal secondo modulo moltiplicativo 209a, sommando fra loro le uscite di secondo strato  $oL2(n)$  e facendo il rappor- 15 to fra le due somme, secondo l'equazione (6).

L'unità di ricostruzione 210 memorizza i pesi di ricostruzione  $oL3(i)$  man mano che questi vengono forniti dall'unità di calcolo dei pesi di ricostruzione 209 e, non appena ha immagazzinato  $2N+1$  pesi di ricostru- 20 zione  $oL3$ , calcola il campione di uscita  $u(2N+1)$ , in base alla (7), utilizzando anche il valore dei  $2N+1$  campioni, forniti dalla memoria dati 200. In seguito, al ricevimento del successivo peso di ricostruzione  $oL3(2N+2)$ , esso calcola il campione di uscita  $u(2N+2)$ , 25 usando anche i  $2N$  pesi di ricostruzione  $oL3$  precedenti



-29-

e altrettanti campioni provenienti dalla memoria dati 200, in accordo con l'equazione (7).

L'unità di controllo 206 determina la sequenza di elaborazione e il caricamento/trasferimento dei dati  
5 fra le diverse unità e moduli. Per calcolare un singolo campione di uscita, l'unità di controllo fa ripetere la sequenza dei passi dell'algoritmo fuzzy per  $2N+1$  volte, aggiorna la memoria dati 200 al termine di  $2N+1$  cicli e controlla il caricamento di successivi  $2N+1$  campioni.

10 I vantaggi ottenibili con il metodo e il filtro descritto sono i seguenti. In primo luogo, essi consentono di ridurre il rumore del segnale di ingresso, sia di tipo bianco, sia di tipo colorato e consentono la separazione di segnali con caratteristiche differenti.  
15 Il filtro preserva i fronti ripidi dei segnali, senza determinare perdite delle caratteristiche del segnale, come evidente dal confronto fra le figure 10a, 10b, 10c. In particolare, la figura 10a è relativa ad un segnale vocale non rumoroso (e precisamente ad un fram-  
20 mento di segnale corrispondente alla vocale "e", con un campionamento a 44,1 KHz e una risoluzione di conversione A/D di 16 bit), la figura 10b è relativa allo stesso segnale di figura 10a, in presenza di rumore bianco, e la figura 10c mostra il risultato del fil-  
25 traggio del segnale di figura 10b con il dispositivo di

-30-

filtro 95 di figura 7.

Dato che il filtro può essere addestrato, esso può essere sia adattato ad uno specifico tipo di segnale iniziale sia modificato successivamente in caso di necessità. Ad esempio, il filtro può essere adattato inizialmente ad un primo tipo di segnale acustico (ad esempio voce maschile con rumore bianco, eseguendo un addestramento con due segnali, uno in presenza di rumore bianco ed uno in assenza di rumore). Quindi, la fase di apprendimento può essere ripetuta per segnali di tipo diverso, ad esempio una voce femminile. I nuovi pesi così calcolati possono sostituire i precedenti, o essere memorizzati in aggiunta; in questo caso l'unità di controllo 206 del filtro 1 può comandare l'invio alle unità 207-209 (figura 9) dei primi o dei secondi campioni a seconda dell'uso del filtro.

Risulta infine chiaro che al metodo e al filtro qui descritto ed illustrato possono essere apportate numerose modifiche e varianti, tutte rientranti nell'ambito del concetto inventivo, come definito nelle rivendicazioni allegate.

- 1 -

## RIVENDICAZIONI

1. Dispositivo di filtraggio (80; 95) di segnali aventi fronti ripidi, caratterizzato dal fatto di comprendere almeno un filtro neuro-fuzzy (1).

5        2. Dispositivo di filtraggio secondo la rivendicazione 1, caratterizzato dal fatto che detto filtro neuro-fuzzy (1) comprende:

un'unità di calcolo caratteristiche di segnale (2), ricevente campioni di ingresso  $e(i)$  di un segnale  
10 da filtrare In e generante caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ );

una rete neuro-fuzzy (3), ricevente dette caratteristiche di segnale e generante pesi di ricostruzione ( $oL3(i)$ ); e

15        un'unità di ricostruzione a media mobile (4), ricevente detti campioni di ingresso  $e(i)$  e detti pesi di ricostruzione ( $oL3(i)$ ) e generante campioni di uscita ( $u(i)$ ) a partire da detti campioni di ingresso e detti pesi di ricostruzione.

20        3. Dispositivo di filtraggio secondo la rivendicazione 2, caratterizzato dal fatto che detta unità di calcolo caratteristiche di segnale (2) comprende una memoria (200) memorizzante una prima pluralità di campioni di ingresso ( $e(i)$ ) formante una prima finestra ed  
25 una rete di calcolo caratteristiche (201, 205) riceven-

- 2 -

te detta prima pluralità di campioni di ingresso e fornente una pluralità di caratteristiche ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) per ogni detto campione di ingresso.

4. Dispositivo secondo la rivendicazione 3, caratterizzato dal fatto che detta rete di calcolo caratteristiche (201, 205) comprende primi mezzi di determinazione caratteristiche (201), generanti, per ogni detto campione di ingresso ( $e(i)$ ), una prima caratteristica di segnale ( $X1(i)$ ) correlata ad una posizione di detto  
10 campione di ingresso in detta prima finestra; secondi mezzi di determinazione caratteristiche (205a), generanti, per ogni detto campione di ingresso, una seconda caratteristica di segnale ( $X2(i)$ ) correlata alla differenza fra detto campione di ingresso ed un campione  
15 centrale ( $e(N)$ ) in detta prima finestra; e terzi mezzi di determinazione caratteristiche (205b), generanti, per ogni detto campione di ingresso, una terza caratteristica di segnale ( $X3(i)$ ) correlata alla differenza fra detto campione di ingresso ed un valore medio di  
20 campione in detta prima finestra.

5. Dispositivo secondo la rivendicazione 4, caratterizzato dal fatto che detti primi mezzi di determinazione caratteristiche (201) generano detta prima caratteristica di segnale ( $X1(i)$ ) per un campione di ingresso  
25 so  $e(i)$  in base alla relazione:

- 3 -

$$X1(i) = \frac{|i-N|}{N}$$

in cui  $i$  è la posizione di detto campione di ingresso  $e(i)$  in detta prima finestra ed  $N$  è la posizione di un campione centrale in detta prima finestra.

5           6. Dispositivo secondo la rivendicazione 5, caratterizzato dal fatto che detti primi mezzi di determinazione caratteristiche comprendono una prima memoria di lavoro (201) memorizzante detta prima caratteristica di segnale ( $X1(i)$ ) per ogni detto campione di ingresso  
10   ( $e(i)$ ).

7. Dispositivo secondo una qualsiasi delle rivendicazioni 4-6, caratterizzato dal fatto che detti secondi mezzi di determinazione caratteristiche (205a) generano detta seconda caratteristica di segnale  
15   ( $X2(i)$ ) per un campione di ingresso  $e(i)$  in base alla relazione:

$$X2(i) = \frac{|e(i) - e(N)|}{\max(diff)}$$

in cui  $e(N)$  è un campione centrale in detta prima finestra e  $\max(diff)$  è il massimo delle differenze fra  
20   tutti i campioni di ingresso in detta prima finestra e detto campione centrale.

8. Dispositivo una qualsiasi delle rivendicazioni 4-7, caratterizzato dal fatto che detti terzi mezzi di

- 4 -

determinazione caratteristiche (205b) generano detta terza caratteristica di segnale ( $X3(i)$ ) per un campione di ingresso  $e(i)$  in base alla relazione:

$$X3(i) = \frac{|e(i) - av|}{\max(diff\_av)}$$

5        in cui  $av$  è il valore medio dei campioni di ingresso in detta prima finestra e  $\max(diff\_av)$  è il massimo delle differenze fra tutti i campioni di ingresso in detta prima finestra e detto valore medio  $av$ .

9. Dispositivo secondo una qualsiasi delle rivendicazioni 2-8, caratterizzato dal fatto che detta rete  
10        neuro-fuzzy (3) comprende:

          neuroni di fuzzificazione (6, 202), ricevanti dette caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) di un campione di ingresso ( $e(i)$ ) e generanti uscite di  
15        primo strato ( $oL1$ ) costituenti il livello di confidenza di dette caratteristiche di segnale rispetto a prefissate funzioni di appartenenza;

          neuroni di fuzzy AND (7), ricevanti dette uscite di primo strato e generanti uscite di secondo strato  
20        ( $oL2$ ) derivanti da regole fuzzy; e

          un neurone di defuzzificazione (8), ricevante dette uscite di secondo strato e generanti un peso di ricostruzione ( $oL3$ ) per ogni detto campione di ingresso ( $e(i)$ ) utilizzando il criterio del centro di gravità.

- 5 -

10. Dispositivo secondo la rivendicazione 9, caratterizzato dal fatto che dette funzioni di appartenenza sono funzioni gaussiane e dette uscite di primo strato sono calcolate in base all'equazione:

$$5 \quad oL1(l,k) = \exp \left( - \left( \frac{Xl - W_m(l,k)}{W_v(l,k)} \right)^2 \right)$$

in cui  $oL1(l,k)$  è un'uscita di primo strato;  $Xl$  è una caratteristica di segnale,  $W_m(l,k)$  è il valore medio e  $W_v(l,k)$  è la varianza di una funzione gaussiana.

11. Dispositivo di filtraggio secondo la rivendicazione 9 o 10, caratterizzato dal fatto di comprendere due funzioni di appartenenza per ogni detta caratteristica di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ).

12. Dispositivo di filtraggio secondo una qualsiasi delle rivendicazioni 9-11, caratterizzato dal fatto che detti neuroni di primo strato (6) comprendono una seconda memoria di lavoro (202) memorizzante valori di dette uscite di primo strato ( $oL1(l,k)$ ) per ogni valore di dette caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ).

13. Dispositivo di filtraggio secondo una qualsiasi delle rivendicazioni 9-12, caratterizzato dal fatto che dette regole fuzzy sono di tipo AND e dette uscite di secondo strato ( $oL2$ ) utilizzano la norma del minimo.

14. Dispositivo di filtraggio secondo la rivendi-

- 6 -

cazione 13, caratterizzato dal fatto che dette uscite di secondo strato sono calcolate in base all'equazione:

$$oL2(n) = \min_n \{W_{FA}(m, n) \cdot oL1(m)\}$$

in cui  $oL2(n)$  è un'uscita di secondo strato,  
 5  $W_{FA}(m, n)$  è un peso di secondo strato e  $oL1(l, k)$  è un'uscita di primo strato.

15. Dispositivo di filtraggio secondo la rivendicazione 14, caratterizzato dal fatto che detti neuroni di secondo strato (7) comprendono una pluralità di unità  
 10 tà moltiplicative (213) ed una pluralità di moduli di minimo (214) collegati in cascata.

16. Dispositivo di filtraggio secondo una qualsiasi delle rivendicazioni 9-15, caratterizzato dal fatto che detti pesi di ricostruzione sono calcolati in base  
 15 all'equazione:

$$oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)}$$

in cui  $oL3$  è un peso di ricostruzione,  $W_{DF}(n)$  sono pesi di terzo strato e  $oL2(n)$  è un'uscita di secondo strato.

20 17. Dispositivo di filtraggio secondo una qualsiasi delle rivendicazioni 9-16, caratterizzato dal fatto che detta unità di ricostruzione a media mobile (4) riceve una seconda pluralità di campioni di ingresso



- 7 -

(e(i)) formante una seconda finestra ed una corrispondente pluralità di pesi di ricostruzione (oL3(i)) e calcola ciascun detto campione di uscita (u(i)) in base alla equazione:

$$5 \quad u(i) = \frac{\sum_{j=0}^{2N} oL3(i-j) \cdot e(i-j)}{\sum_{j=0}^{2N} e(i-j)}$$

in cui e(i-j) è un campione di ingresso (i-j)-esimo e oL3(i-j) è un peso di ricostruzione associato ad un campione di ingresso (i-j)-esimo.

18. Dispositivo di filtraggio secondo una qualsiasi  
 10 si delle rivendicazioni 9-17, caratterizzato dal fatto di comprendere una unità di addestramento (5) avente un primo ingresso collegato a detta unità di ricostruzione a media mobile (4) e ricevente detti campioni di uscita (u(i)), un secondo ingresso ricevente un segnale di  
 15 uscita desiderato (Tg) ed un'uscita collegata a detta rete neuro-fuzzy (3) per fornire valori di pesatura ottimizzati.

19. Dispositivo di filtraggio secondo una qualsiasi  
 si delle rivendicazioni 1-18, caratterizzato dal fatto  
 20 di comprendere almeno un primo stadio di divisione ("splitting stage") (H<sub>1</sub>, H<sub>0</sub>, 84, 85; H<sub>001</sub>, H<sub>102</sub>, 96, 97) generante almeno due flussi di campioni da filtrare; un detto filtro neuro-fuzzy (86, 87; 120-127) per ogni

- 8 -

flusso di campioni da filtrare, detti filtri neuro-fuzzy generando ciascuno un rispettivo flusso di campioni filtrati; ed un primo stadio di ricomposizione (88, 89, 92,  $G_0$ ,  $G_1$ ; 154, 155, 160,  $G_{001}$ ,  $G_{102}$ ) ricevente  
5 detti flussi di campioni filtrati e generante un unico flusso di campioni di uscita ( $s_0$ ).

20. Dispositivo di filtraggio secondo la rivendicazione 19, caratterizzato dal fatto di comprendere inoltre una pluralità di ulteriori stadi di divisione  
10 ( $H_{011}$ - $H_{128}$ , 100-114), disposti in cascata fra loro e a detto primo stadio di divisione ( $H_{001}$ ,  $H_{102}$ , 96, 97) ed una pluralità di ulteriori stadi di ricomposizione (130-151,  $G_{021}$ - $G_{114}$ ), disposti in cascata fra loro, fra detti filtri neuro-fuzzy (120-127) e detto primo stadio  
15 di ricomposizione (154, 155, 160,  $G_{001}$ ,  $G_{102}$ ).

21. Dispositivo di filtraggio secondo la rivendicazione 19 o 20, caratterizzato dal fatto che detti stadi di divisione ( $H_1$ ,  $H_0$ , 84, 85;  $H_{001}$ - $H_{128}$ , 96-114) comprendono ciascuno un primo ed un secondo filtro di analisi ( $H_1$ ,  $H_0$ ;  $H_{001}$ - $H_{128}$ ) fra loro in quadratura, ricevanti  
20 un flusso di campioni da dividere e generanti ciascuno un rispettivo flusso di campioni divisi e una prima ed una seconda unità a riduttore di campioni ("downsampler unit") (84, 85; 96-114) ricevente ciascuna un rispetti-  
25 vo detto flusso di campioni divisi,

- 9 -

e dal fatto che detti stadi di ricomposizione (88, 89, 92,  $G_0$ ,  $G_1$ ; 130-160,  $G_{021}$ - $G_{102}$ ) comprendono ciascuno una prima ed una seconda unità a incremento di campioni ("upsampler unit") (88, 89; 130-137, 146-149, 154, 5 155), riceventi ciascuna un rispettivo flusso di campioni da incrementare e generando un rispettivo flusso di campioni incrementati; un primo ed un secondo filtro di sintesi ( $G_0$ ,  $G_1$ ;  $G_{021}$ - $G_{102}$ ) fra loro in quadratura e complementari a detti filtri di analisi, riceventi cia- 10 scuno un rispettivo flusso di campioni incrementati e generando ciascuno un rispettivo flusso di campioni parziali; ed un nodo sommatore (92; 140-143, 150, 151, 160), ricevente detti flussi di campioni parziali e generante un flusso di campioni sommati.

15        22. Dispositivo di filtraggio secondo la rivendicazione 21, caratterizzato dal fatto che detti filtri di analisi ( $H_1$ ,  $H_0$ ,  $H_{001}$ - $H_{128}$ ) sono filtri QMF e detti filtri di sintesi ( $G_0$ ,  $G_1$ ;  $G_{021}$ - $G_{102}$ ) sono filtri QMF complementari a detti filtri di analisi.

20        23. Dispositivo di filtraggio secondo la rivendicazione 22, caratterizzato dal fatto che detti filtri QMF ( $H_1$ ,  $H_0$ ,  $H_{001}$ - $H_{128}$ ;  $G_0$ ,  $G_1$ ;  $G_{021}$ - $G_{102}$ ) sono filtri convolutivi.

25        24. Metodo per ridurre il rumore in un filtro dotato di fronti ripidi, caratterizzato dal fatto di ese-

- 10 -

guire una fase di filtrazione neuro-fuzzy.

25. Metodo secondo la rivendicazione 24, caratterizzato dal fatto di comprendere le fasi di:

calcolare caratteristiche di segnale ( $X1(i)$ ,  
5  $X2(i)$ ,  $X3(i)$ ) a partire da campioni di ingresso ( $e(i)$ )  
di un segnale da filtrare ( $In$ );

calcolare pesi di ricostruzione ( $oL3$ ) a partire da  
dette caratteristiche di segnale tramite una rete neuro-fuzzy (3); e

10 ricostruire a media mobile un segnale di uscita  
( $Out$ ) formato da una pluralità di campioni di uscita  
( $u(i)$ ) a partire da detti campioni di ingresso e detti  
pesi di ricostruzione.

26. Metodo secondo la rivendicazione 25, caratterizzato dal fatto di comprendere le fasi di:

15 memorizzare una prima pluralità di campioni di ingresso ( $e(i)$ ) formante una prima finestra;

calcolare una pluralità di caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) per ogni detto campione di  
20 ingresso a partire da detta prima pluralità di campioni di ingresso.

27. Metodo secondo la rivendicazione 26, caratterizzato dal fatto che detta fase di calcolare una pluralità di caratteristiche di segnale per ogni detto  
25 campione di ingresso comprende le fasi di:

- 11 -

calcolare una prima caratteristica di segnale ( $X1(i)$ ) correlata ad una posizione di detto campione di ingresso ( $e(i)$ ) in detta prima finestra;

calcolare una seconda caratteristica di segnale  
5 ( $X2(i)$ ) correlata alla differenza fra detto campione di ingresso ed un campione centrale in detta prima finestra; e

calcolare una terza caratteristica di segnale ( $X3(i)$ ) correlata alla differenza fra detto campione di  
10 ingresso ed un valore medio di campione ( $av$ ) in detta prima finestra.

28. Metodo secondo la rivendicazione 26 o 27, caratterizzato dal fatto che detta fase di calcolare pesi di ricostruzione ( $oL3$ ) comprende le fasi di:

15 eseguire un'operazione di fuzzificazione e calcolare uscite di primo strato ( $oL1$ ) costituenti il livello di confidenza di dette caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) rispetto a prefissate funzioni di appartenenza;

20 eseguire un'operazione di fuzzy AND e generare uscite di secondo strato ( $oL2$ ) derivanti da regole fuzzy a partire da dette uscite di primo strato; e

eseguire un'operazione di defuzzificazione su dette uscite di secondo strato e generare un peso di ricostruzione ( $oL3(i)$ ) per ogni detto campione di ingresso  
25

- 12 -

(e(i)) utilizzando il criterio del centro di gravità.

29. Metodo secondo la rivendicazione 28, caratterizzato dal fatto che dette funzioni di appartenenza sono funzioni gaussiane e dette uscite di primo strato  
5 sono calcolate in base all'equazione:

$$oL1(l,k) = \exp \left( - \left( \frac{Xl - W_m(l,k)}{W_v(l,k)} \right)^2 \right)$$

in cui  $oL1(l,k)$  è un'uscita di primo strato;  $Xl$  è una caratteristica di segnale,  $W_m(l,k)$  è il valore medio e  $W_v(l,k)$  è la varianza di una funzione gaussiana.

10 30. Metodo secondo la rivendicazione 28 o 29, caratterizzato dal fatto che dette uscite di secondo strato sono calcolate in base all'equazione:

$$oL2(n) = \min_n \{ W_{FA}(m,n) \cdot oL1(m) \}$$

in cui  $oL2(n)$  è un'uscita di secondo strato,  
15  $W_{FA}(m,n)$  è un peso di secondo strato e  $oL1(l,k)$  è un'uscita di primo strato.

31. Metodo secondo una qualsiasi delle rivendicazioni 28-30, caratterizzato dal fatto che detti pesi di ricostruzione sono calcolati in base all'equazione:

$$20 \quad oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)}$$

in cui  $oL3$  è un peso di ricostruzione,  $W_{DF}(n)$  sono pesi di terzo strato e  $oL2(n)$  è un'uscita di secondo

- 13 -

strato.

32. Metodo secondo una qualsiasi delle rivendicazioni 26-31, caratterizzato dal fatto che detta fase di ricostruire a media mobile comprende le fasi di:

- 5 ricevere una seconda pluralità di campioni di ingresso ( $e(i)$ ) formante una seconda finestra ed una corrispondente pluralità di pesi di ricostruzione ( $oL3(i)$ ) e calcolare ciascun detto campione di uscita in base alla equazione:

$$10 \quad u(i) = \frac{\sum_{j=0}^{2N} oL3(i-j) \cdot e(i-j)}{\sum_{j=0}^{2N} e(i-j)}$$

in cui  $e(i-j)$  è campione di ingresso  $(i-j)$ -esimo e  $oL3(i-j)$  è un peso di ricostruzione associato ad un campione di ingresso  $(i-j)$ -esimo.

- 15 33. Metodo secondo una qualsiasi delle rivendicazioni 24-32, caratterizzato dal fatto di effettuare una fase di addestramento di pesi utilizzati in detta fase di filtrazione neuro-fuzzy.

- 20 34. Metodo secondo la rivendicazione 33, caratterizzato dal fatto che detta fase di addestramento comprende le fasi di:

generare un segnale di ingresso di configurazione nota;

filtrare detto segnale di ingresso di configura-

- 14 -

zione nota per ottenere un segnale di uscita di prova;

confrontare detto segnale di uscita di prova con un segnale desiderato ( $T_g$ ) per ottenere una distanza fra detto segnale di uscita di prova e un segnale desi-

5 derato;

calcolare una funzione di conformità ("fitness") a partire da detta distanza;

ottimizzare detti pesi in base a detta funzione di conformità.

10 35. Metodo secondo una qualsiasi delle rivendicazioni 26-34, caratterizzato dal fatto di effettuare una analisi multirisoluzione con suddivisione di segnale in sottobande tramite ondine ortonormali.

36. Metodo secondo la rivendicazione 35, caratterizzato dal fatto di comprendere le fasi di:

15 dividere un flusso di campioni di ingresso in almeno due flussi di campioni da filtrare;

filtrare ogni flusso di campioni da filtrare tramite un rispettivo filtro neuro-fuzzy (86, 87; 120-127)

20 per ottenere almeno due flussi di campioni filtrati; e

ricomporre detti flussi di campioni filtrati per generare un unico flusso di campioni di uscita ( $s_0$ ).

37. Metodo secondo la rivendicazione 36, caratterizzato dal fatto che, prima di eseguire detta fase di

25 filtrare, detta fase di dividere viene ripetuta un nu-



- 15 -

mero prefissato di volte e che, dopo aver eseguito detta fase di filtrare, detta fase di ricomporre viene ripetuta detto numero prefissato di volte.

38. Metodo secondo la rivendicazione 36 o 37, caratterizzato dal fatto che detta fase di dividere comprende le fasi di:

alimentare un flusso di campioni da dividere a due filtri di analisi fra loro in quadratura;

generare due flussi di campioni divisi filtrati; e  
10 ridurre in campioni ("downsampling") detti flussi di campioni divisi filtrati,

e dal fatto che detta fase di ricomporre comprende le fasi di:

incrementare flussi di campioni da incrementare,  
15 generando flussi di campioni incrementati;

filtrare detti flussi di campioni incrementati tramite due filtri di sintesi fra loro in quadratura e complementari a detti filtri di analisi, generando flussi di campioni parziali; e

20 sommare coppie di flussi di campioni parziali e generare un flusso di campioni sommati.

- 1 -

## RIASSUNTO

Il dispositivo di filtro (80) comprende un filtro neuro-fuzzy (1; 80) ed implementa una tecnica di filtraggio a media mobile in cui i pesi per la ricostruzione  
5 finale del segnale ( $oL3(i)$ ) sono calcolati in una rete neuro-fuzzy (3) in base a specifiche regole fuzzy. Le regole fuzzy operano su tre caratteristiche di segnale ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) per ogni campione di ingresso ( $e(i)$ ). Le caratteristiche di segnale sono correlate  
10 alla posizione del campione nella finestra di campioni considerata, alla differenza fra un campione e il campione al centro della finestra e alla differenza tra un campione e la media di campioni nella finestra. Il dispositivo di filtro per l'analisi di un segnale vocale  
15 comprende un banco di filtri neuro-fuzzy (86, 87). Il segnale viene separato in diverse sotto-bande, secondo la teoria delle ondine, utilizzando un banco di filtri di analisi, includente una coppia di filtri FIR QMF ( $H_0$ ,  $H_1$ ) ed una coppia di riduttori di campioni (85, 86);  
20 ciascun segnale di sotto-banda viene filtrato da un filtro neuro-fuzzy (86, 87) e quindi le varie sotto-bande vengono ricostruite da un banco di filtri di sintesi, includente una coppia di incrementatori di campioni (88, 89), una coppia di filtri FIR QMF ( $G_0$ ,  $G_1$ ) ed  
25 un nodo sommatore (92). Figure 1 e 4

1 / 8

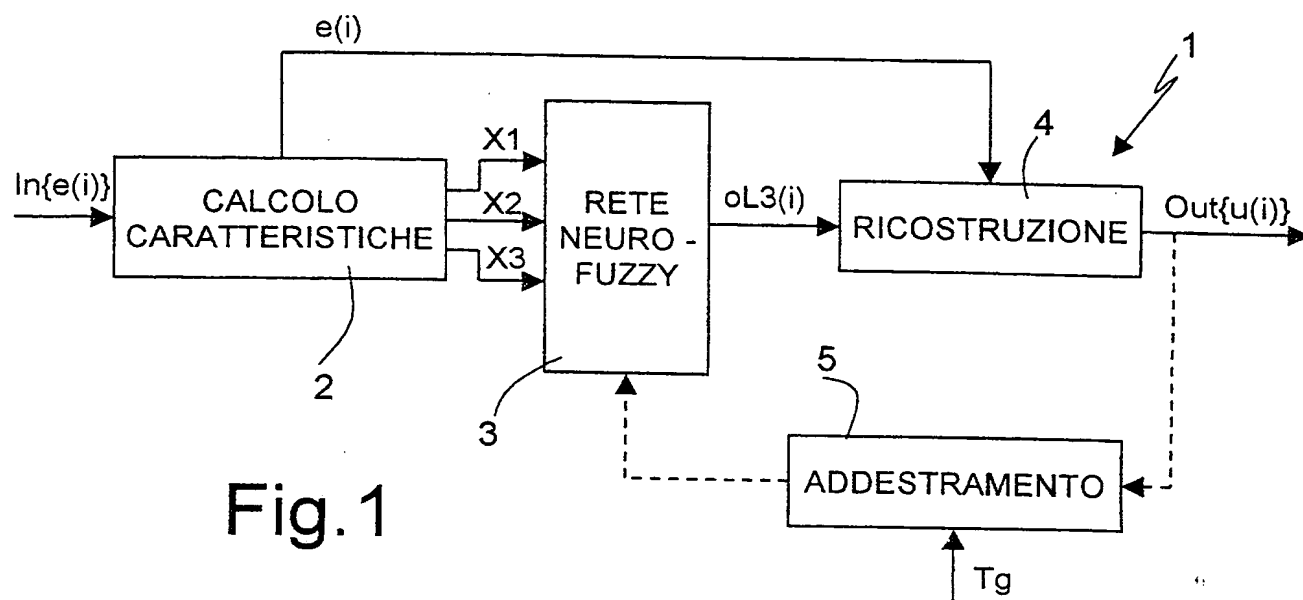


Fig.1

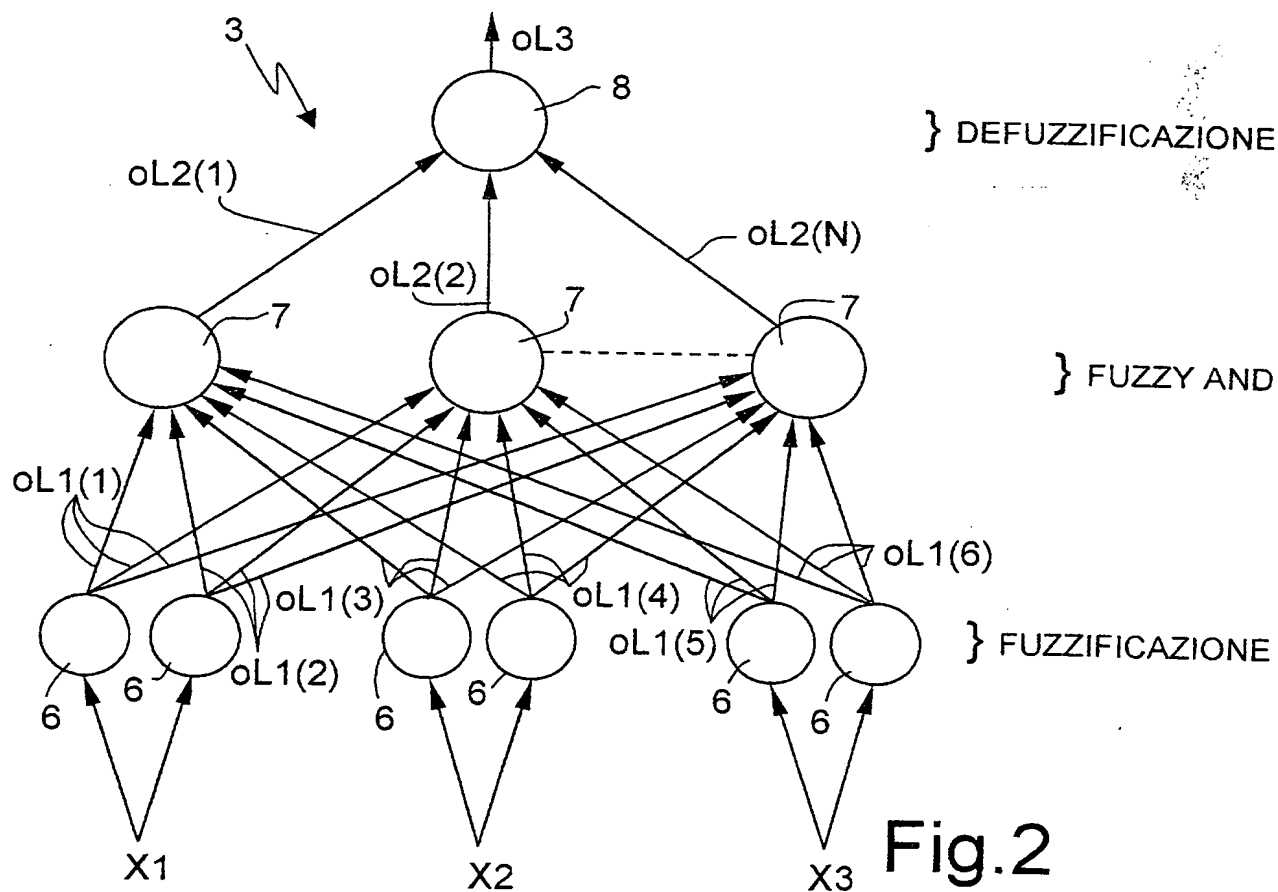


Fig.2

2 / 8

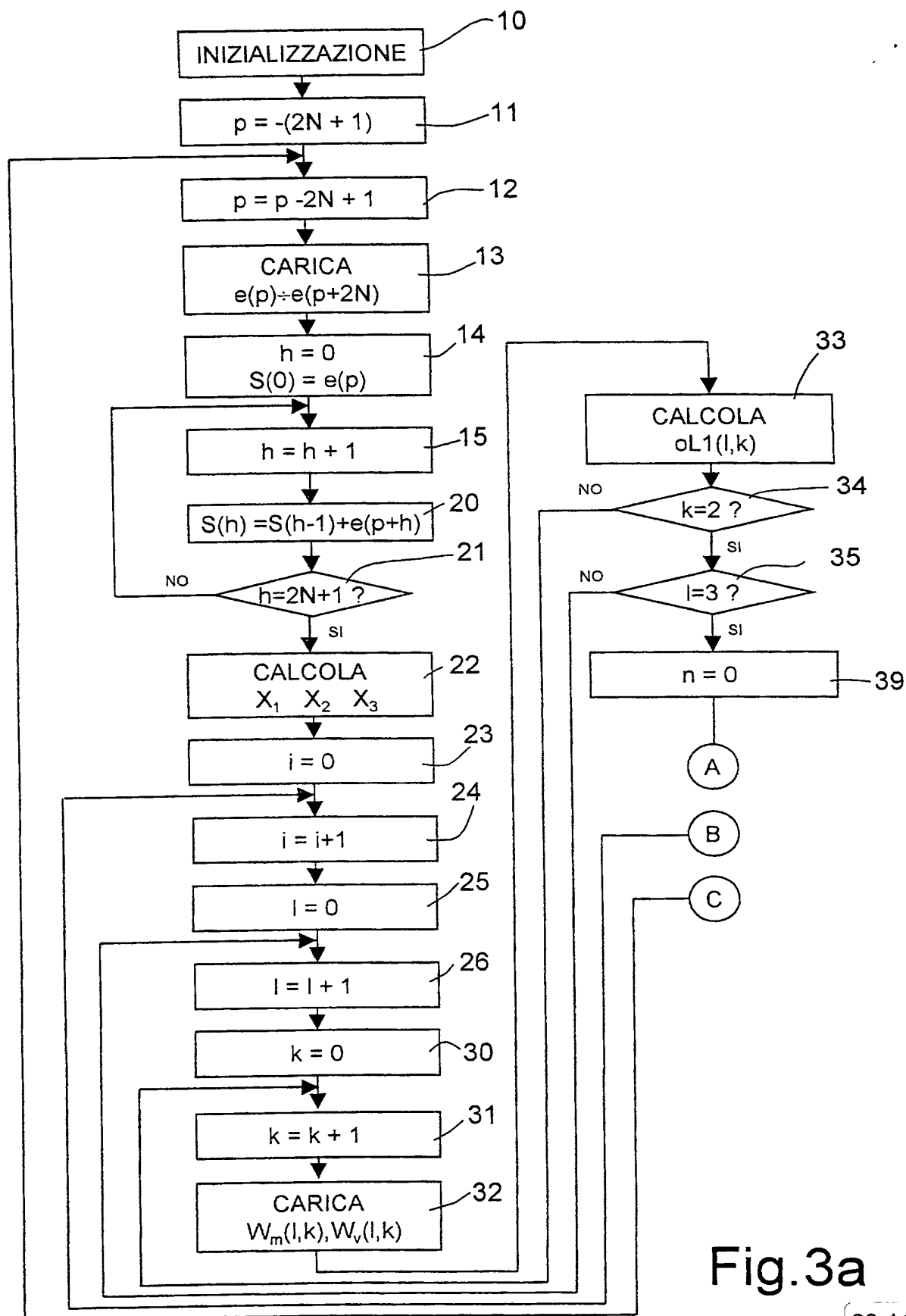


Fig.3a

3 / 8

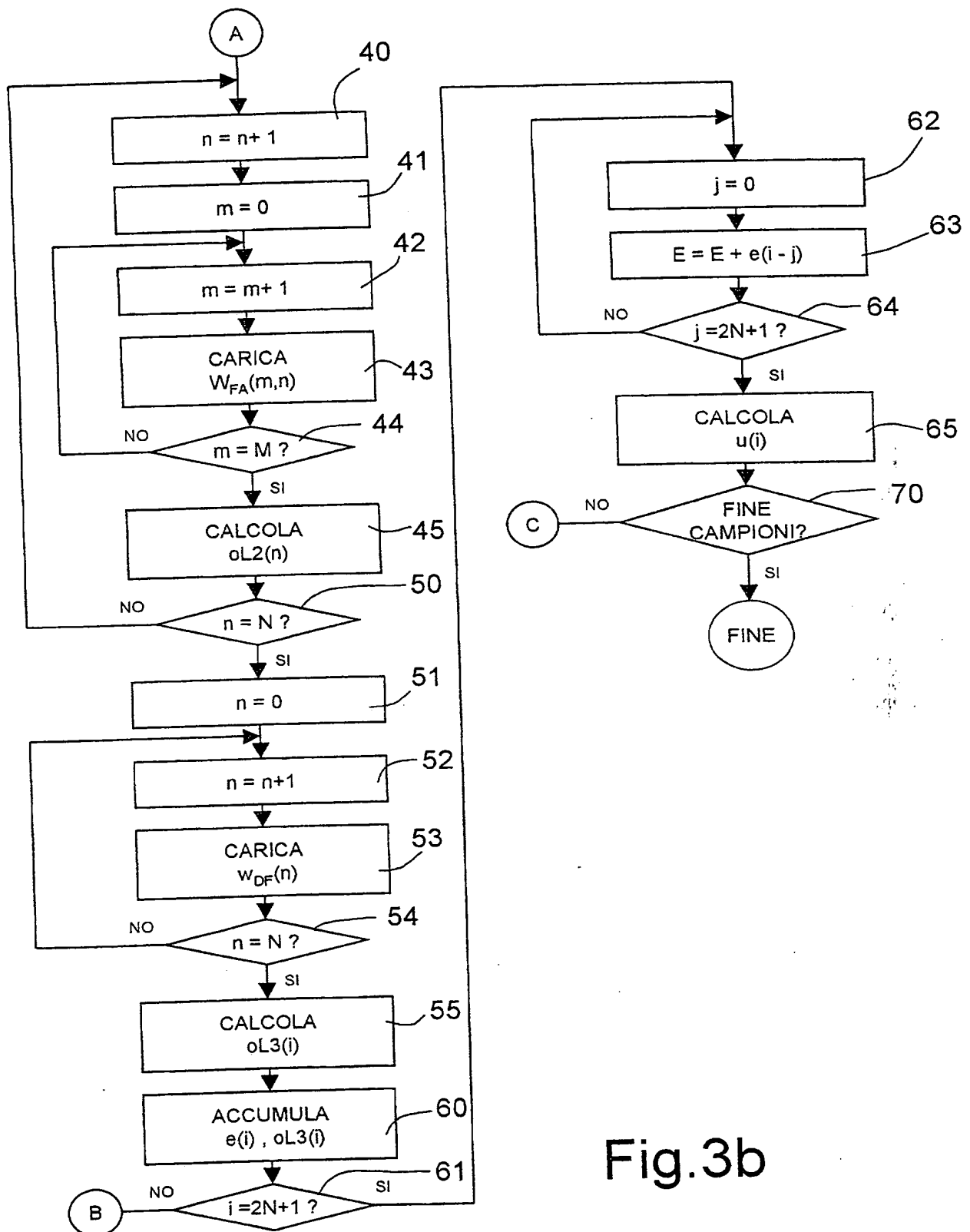


Fig.3b

4 / 8

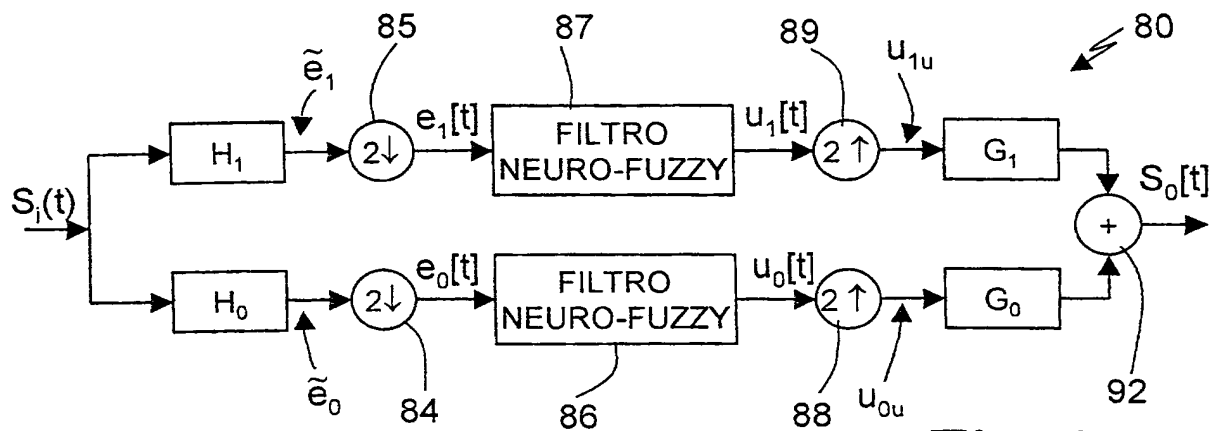


Fig.4

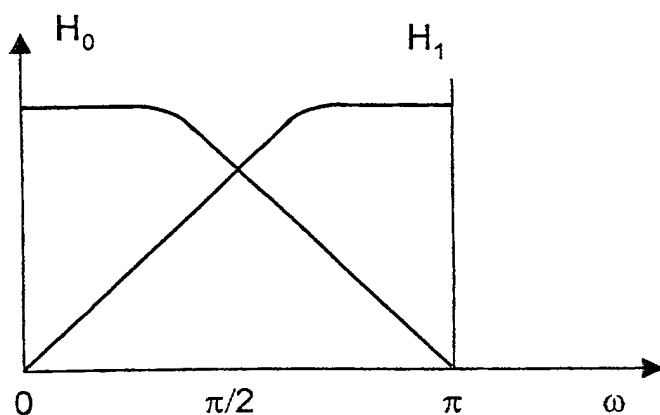


Fig.5

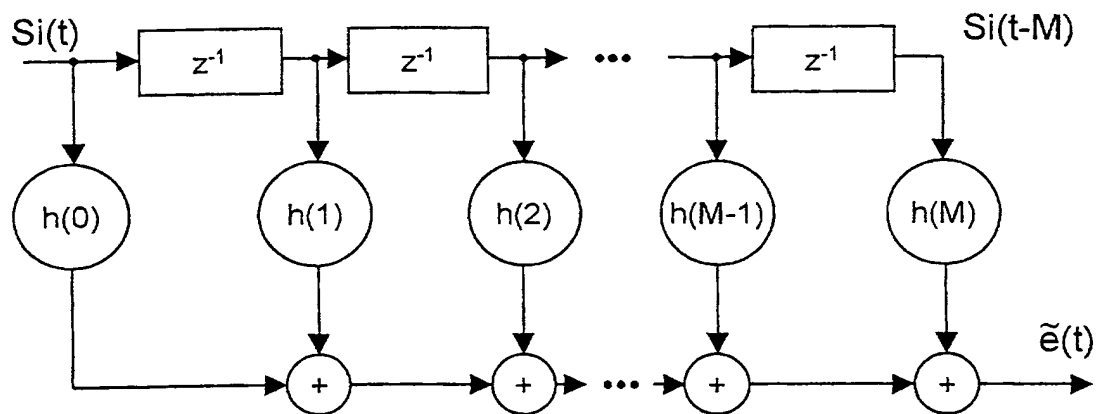


Fig.6

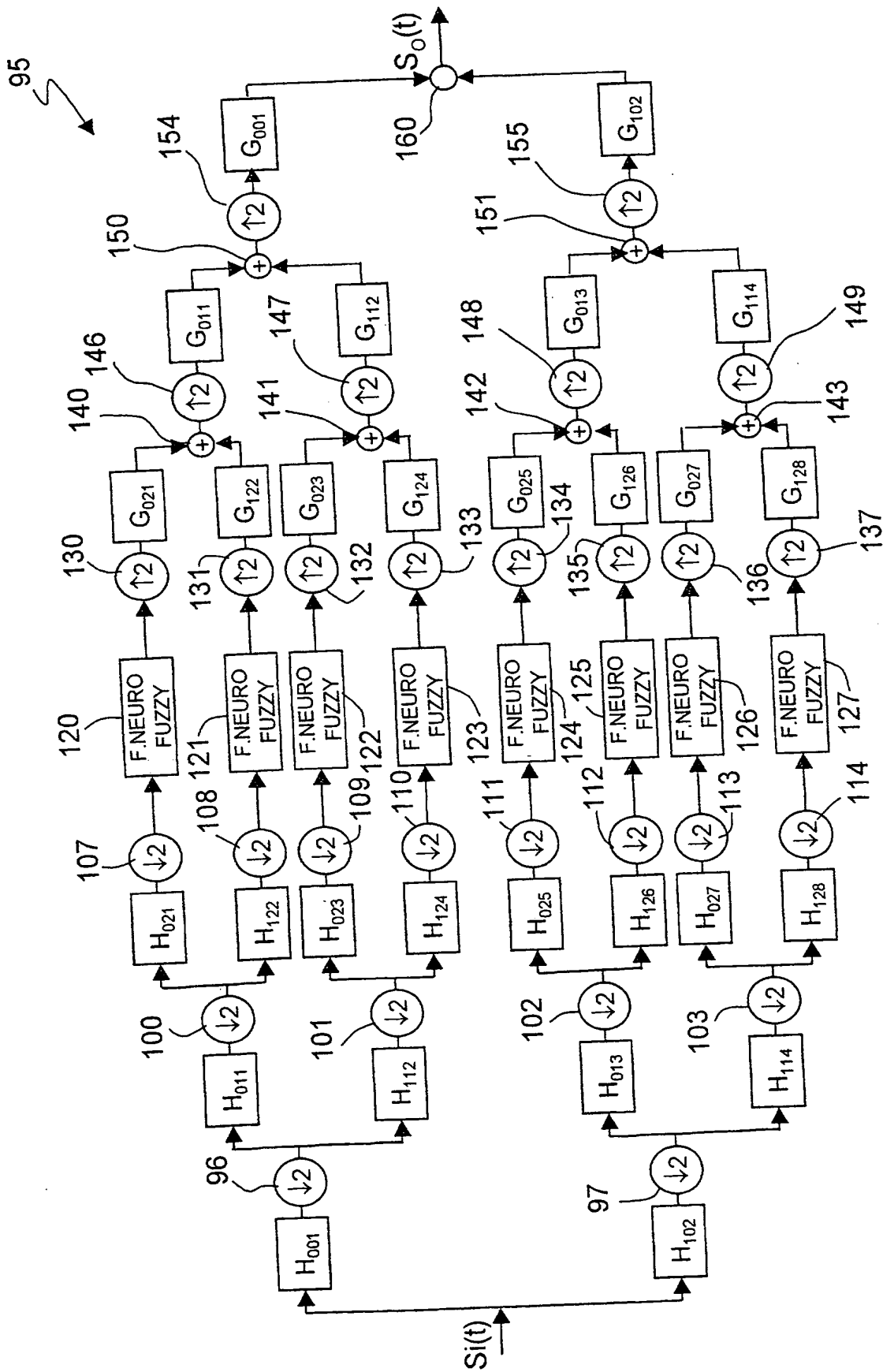


Fig.7

6 / 8

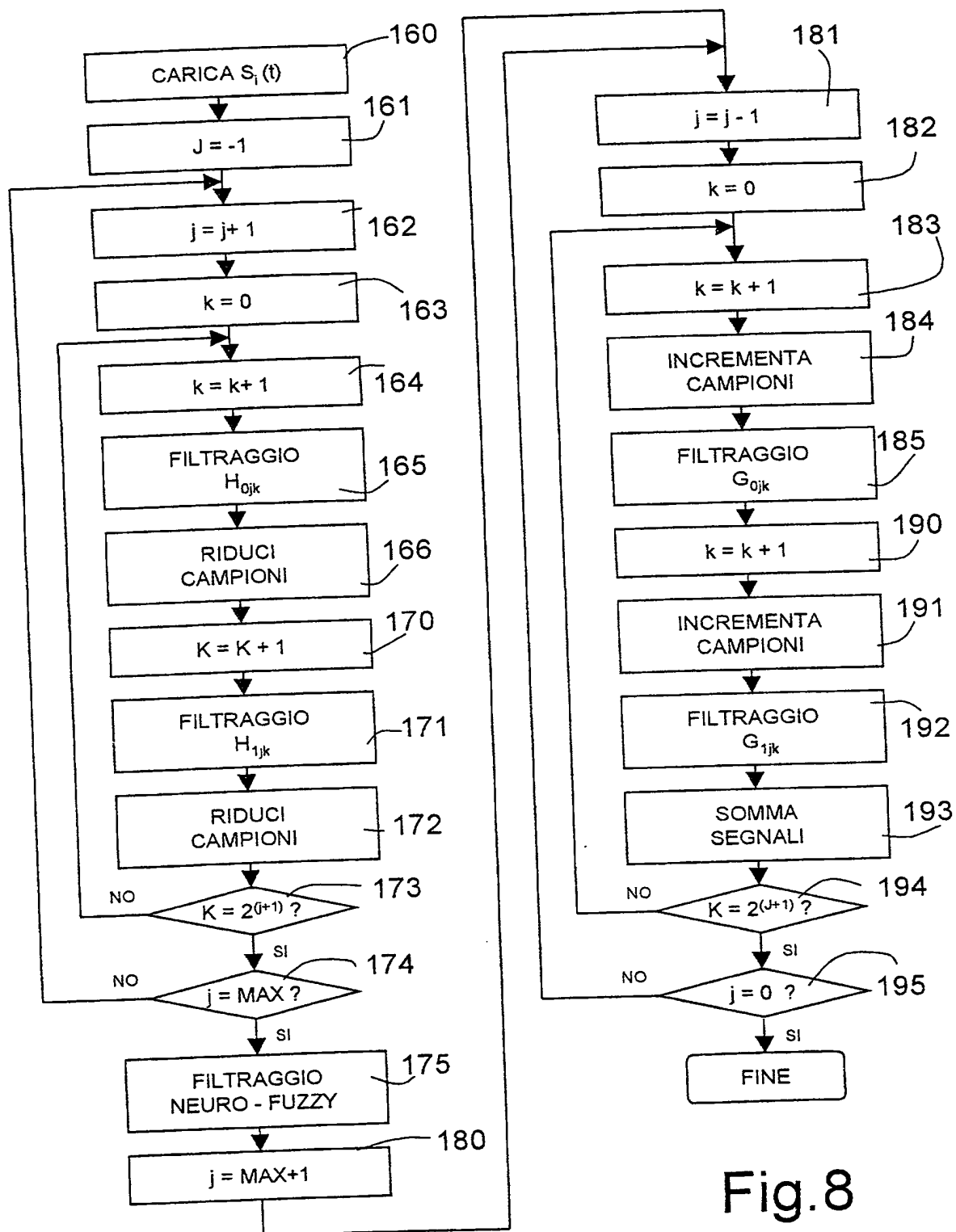


Fig.8



7 / 8

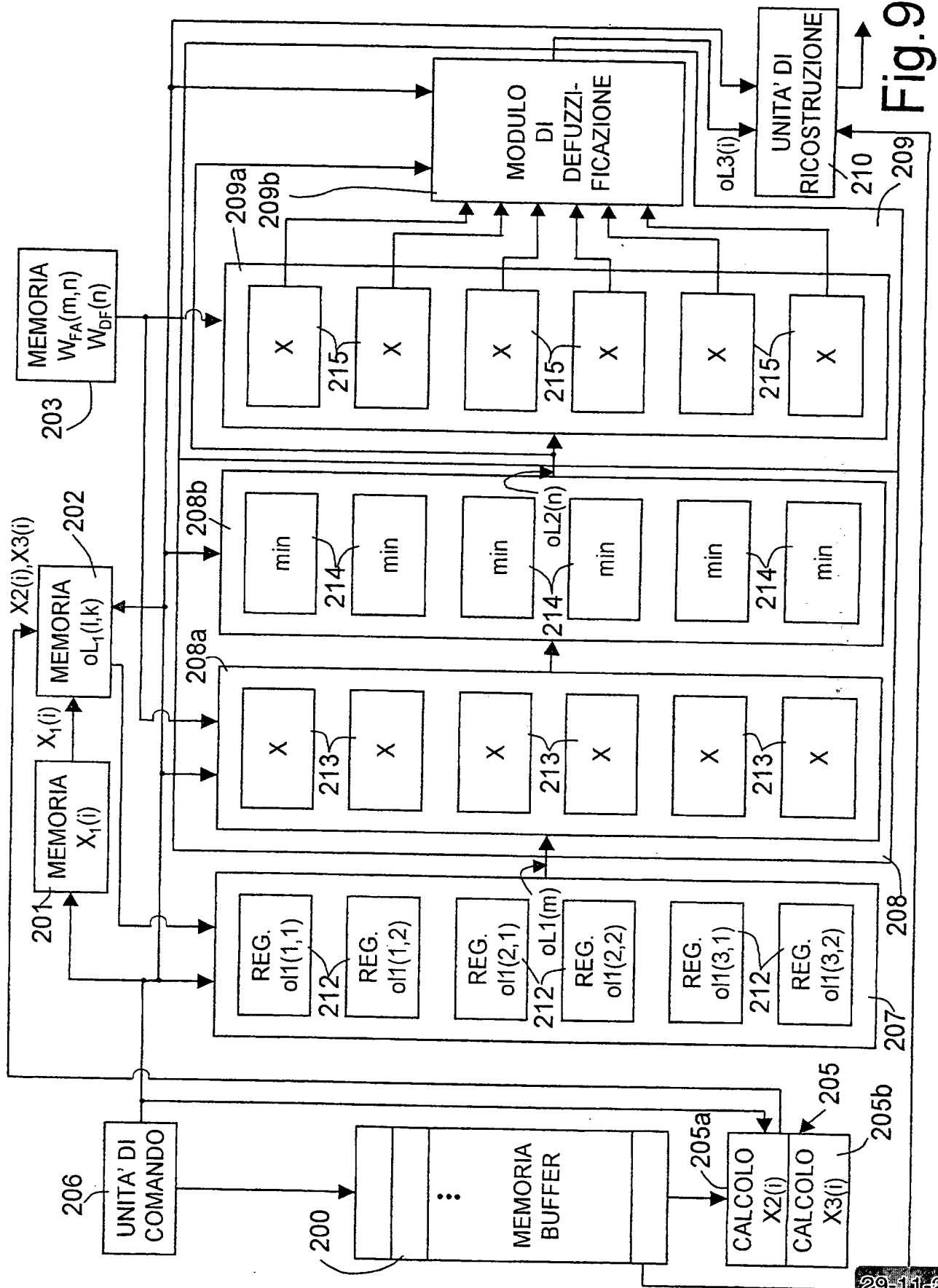


Fig.9

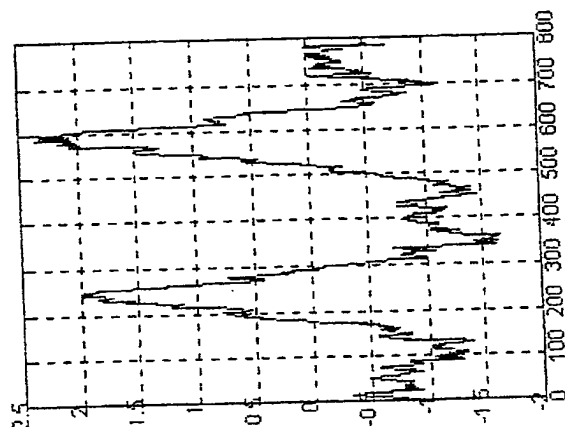


Fig. 10c

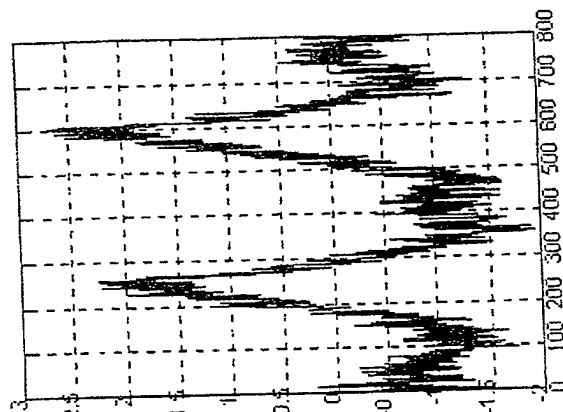


Fig. 10b

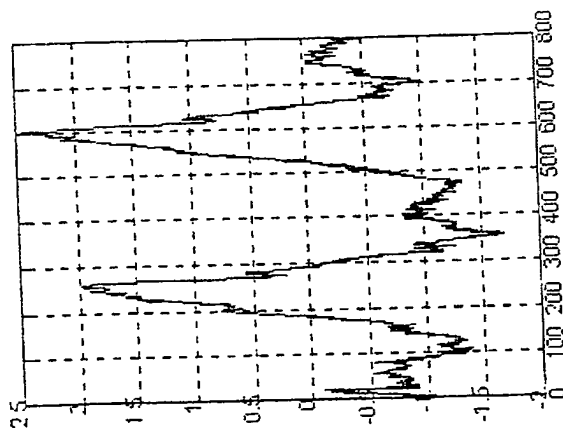


Fig. 10a

22. 01. 2001

- 1 -

(40)

FILTERING DEVICE AND METHOD FOR REDUCING NOISE IN ELECTRICAL SIGNALS, IN PARTICULAR ACOUSTIC SIGNALS AND IMAGES

5 The present invention refers to a filtering device and method for reducing noise in electrical signals, in particular acoustic (voice) signals and images.

10 As is known, filters operating according to various linear and nonlinear techniques are used to remove undesired components from electrical signals. In particular, undesired components may be any type of noise (white noise, flicker, etc.) or other types of superimposed acoustic or visual signals.

15 Linear filters are at present the most widely used instruments for filtering noise. Finite-impulse filters (FIRs) eliminate all the harmonics of a signal having a frequency higher than the cutoff frequency of the filter and improve the signal-to-noise ratio (SNR). Another linear  
20 filtering technique is based on the fast Fourier transform (FFT), where the signal is transformed into the frequency domain, the undesired harmonics are removed, and then the inverse Fourier transform is calculated.

25 As far as nonlinear techniques are concerned, average filters are simple to design and may be implemented through simple hardware circuits. Average filters are based on the comparison of the individual signal samples in an time interval with the average of all the samples in the same  
30 time interval. On the basis of this comparison, the individual samples are selectively attenuated.

35 All these methods share the disadvantage that, when removing the noise, also some of the components of the original signal are removed.

Furthermore, none of the current techniques, whether linear or nonlinear ones, including average filtering, is able to preserve steep edges of the signal. If a moving-average filter is used, the width of the window must be very small if steep edges are to be preserved. However, if the size of the window becomes small, there is no significant reduction in noise energy. If linear filters are used, all the frequencies above the cutoff frequency are eliminated, with consequent marked distortion of the signal.

The aim of the invention is thus to provide a filtering method and device that does not cause a sensible deterioration of the signal and at the same time preserves the edges of the signal.

According to the present invention a device and method are provided for filtering electrical signals, as respectively defined in Claim 1 and Claim 24.

The filter and the method described are particularly useful in the case of signals having steep edges, for which the aim is to preserve the edges of the signals. Furthermore, it is possible to filter signals affected by white and non-white noise, such as flicker noise. Through the present method it is moreover possible to eliminate from a signal other signals that are superimposed on it and are characterized by a wide spectral range.

The device and the method described are based upon a neuro-fuzzy network. They are implemented with a moving-average filtering technique in which the weighting factors (or weights) for the final reconstruction of the signal are calculated in a neuro-fuzzy network according to specific fuzzy rules. This enables a better reduction of the noise.

The fuzzy rules operate on different variables, referred to as signal features. Described hereinafter are three signal

features and six fuzzy rules.

The proposed filter is suitable for visual signals or acoustic signals, even ones with sudden variations. Various types of functions or signal features can be used to create the rules. With the method described, the signal features are correlated to the position of the sample in the considered sample window, to the difference between a given sample and the sample at the center of the window, and to the difference between a given sample and the average of samples in the window. These signal features may have a considerable influence on the values of the weights for the reconstruction of the signal; in addition, they may be calculated in a relatively simple manner.

The method and the filter according to the invention moreover comprise a neuro-fuzzy filter bank. In this way, the signal may be split into different sub-bands according to wavelet theory: each sub-band signal may be filtered by a neuro-fuzzy network, and then the various sub-bands can be reconstructed by the synthesis filter bank. As is known from wavelet theory, in the first sub-band the signal features have a low frequency, whereas in the last sub-band the signal features have the maximum frequency. If non-white noise (colored noise) is to be removed, this is approximated by white noise in each individual sub-band. Given that a neuro-fuzzy network works well on white noise, this solution leads to a sensible reduction in noise.

The network is trained by supplying some configurations of input and output signals (the configuration of the output signal that it is required to obtain as a result of the network evolution is called target configuration). The training algorithm is based upon one of the known learning methods, such as gradient descent, a genetic algorithm, the simulated annealing method, random search, or any other

method for function optimization.

For an understanding of the present invention, preferred embodiments thereof are now described, purely to furnish  
5 non-limiting examples, with reference to the attached drawings, in which:

- Figure 1 is a block diagram showing the general architecture of the filter according to the invention;

10 - Figure 2 represents the topology of a block of Figure 1 for a neuro-fuzzy network;

- Figures 3a and 3b show a flowchart of the operations performed by the filter of Figure 1;

- Figure 4 shows a block diagram of a filtering device using the filter of Figure 1;

15 - Figure 5 shows the transfer functions of digital filters belonging to the filtering device of Figure 4;

- Figure 6 shows the block diagram of some digital filters belonging to the filtering device of Figure 4;

20 - Figure 7 shows a block diagram of another filtering device using the filter of Figure 1;

- Figure 8 shows a flowchart of the operation of the filtering device of Figure 7;

- Figure 9 shows a block diagram for the hardware implementation of the filter of Figure 1; and

25 - Figures 10a, 10b, and 10c respectively show a voice signal free from noise, the same signal superimposed on white noise, and the same signal after filtering according to the invention.

30 Figure 1 shows a filter 1 comprising a signal-feature computation unit 2, a neuro-fuzzy network 3, a reconstruction unit 4, and a training unit 5.

35 The signal-feature computation unit 2 receives at the input a signal  $In$  including of a plurality of input samples  $e(i)$ , stores, at each clock cycle,  $(2N + 1)$  input samples  $e(i)$

(which represent a work window for filter 1) in an internal buffer, computes the signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$  for each input sample  $e(i)$  on the basis of all the input samples belonging to the work window (as described in detail hereinafter), and supplies the signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$  thus calculated to the neuro-fuzzy network 3.

The neuro-fuzzy network 3 processes the signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$  and generates at the output a reconstruction weight  $oL3(i)$  for each input sample  $e(i)$ . To this aim, starting from the signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$  and for each input sample  $e(i)$ , the neuro-fuzzy network 3 first performs a fuzzification operation, then applies preset fuzzy rules, and finally carries out a defuzzification operation. The reconstruction weight  $oL3(i)$  thus obtained is hence the weighted sum of all the input samples  $e(i)$  in the same work window, as explained in detail hereinafter with reference to Figures 2 and 3.

The reconstruction unit 4 receives the reconstruction weights  $oL3(i)$  and the input samples  $e(i)$  and, after accumulating a sufficient number of input samples  $e(i)$  and of corresponding reconstruction weights  $oL3(i)$ , generates an output sample  $u(i)$  the sequence of which forms an output signal Out.

The training unit 5 is operative only initially, so as to train the neuro-fuzzy network 3 and modify the weights of the network with the aim of obtaining an optimal behavior of the filter 1, as described in detail hereinafter.

The signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$ , computed in the signal-feature computation unit 2, are correlated, respectively, to the distance between each sample and the central sample of the considered window, to the difference

between a given sample and the sample at the center of the window, and to the difference between a given sample and the average of samples in the window, and are normalized so as to obtain values between 0 and 1.

5

In detail, given a window of  $(2N + 1)$  input samples  $e(i)$ , with  $i = 0, \dots, 2N$ , the signal features  $X1(i)$ ,  $X2(i)$ , and  $X3(i)$  for each input sample  $e(i)$  are defined as

$$X1(i) = \frac{|i - N|}{N} \quad (1)$$

10

$$X2(i) = \frac{|e(i) - e(N)|}{\max(diff)} \quad (2)$$

$$X3(i) = \frac{|e(i) - av|}{\max(diff\_av)} \quad (3)$$

where  $N$  is the position of a central sample  $e(N)$  in the work window;

15

$\max(diff) = \max(e(k) - e(N))$  with  $k = 0, \dots, 2N$ , i.e., the maximum of the differences between all the input samples  $e(k)$  and the central sample  $e(N)$ ;

$av$  is the average value of the input samples  $e(i)$ ; and

20

$\max(diff\_av) = \max(e(k) - av)$  with  $k = 0, \dots, 2N$ , i.e., the maximum of the differences between all the input samples  $e(k)$  and the average value  $av$ .

25

The neuro-fuzzy network 3 is a three-layer fuzzy network the functional representation of which appears in Figure 2, in which, for reasons of simplicity, the index  $i$  in parenthesis for the specific sample within the respective work window is not indicated. Nevertheless, as mentioned above, and as will emerge more clearly from the flowchart of Figures 3a, 3b, the neuro-fuzzy processing represented by Figure 2 is repeated for each input sample  $e(i)$ .

30

In detail, starting from the three signal features  $X1$ ,  $X2$  and  $X3$  (or generically from 1 signal features  $X1$ ) and given  $k$  membership functions of a gaussian type for each signal feature (described by the average value  $W_m(1, k)$  and by the



variance  $W_v(l, k)$ , a fuzzification operation is performed in which the membership level of the signal features  $X_1$ ,  $X_2$  and  $X_3$  is evaluated with respect to each membership function (here two for each signal feature, so that  $k = 2$ ; in all there are  $M = 1 \times k = 6$  membership functions).

In Figure 2, the above operation is represented by six first-layer neurons 6, each of which, starting from the three signal features  $X_1$ ,  $X_2$  and  $X_3$  (generically designated by  $X_l$ ) and using as weights the average value  $W_m(l, k)$  and the variance  $W_v(l, k)$  of the membership functions, supplies a first-layer output  $oL1(l, k)$  (hereinafter also designated by  $oL1(m)$ ) calculated as follows:

$$oL1(l, k) = oL1(m) = \exp \left( - \left( \frac{X_l - W_m(l, k)}{W_v(l, k)} \right)^2 \right) \quad (4)$$

Hereinafter, a fuzzy AND operation is performed, using the norm of the minimum, in such a way as to obtain  $N$  second-layer outputs  $oL2(n)$ . For example,  $N$  is equal to 6. As is known, a fuzzy AND operation using the norm of the minimum is based upon rules of the type:

if  $X_1^{(1)}$  is  $A_1^{(1)}$  and  $X_2^{(1)}$  is  $A_2^{(1)}$  and  $X_3^{(1)}$  is  $A_3^{(1)}$  then  $O$  is  $B_1^{(1)}$

...

if  $X_1^{(n)}$  is  $A_1^{(n)}$  and  $X_2^{(n)}$  is  $A_2^{(n)}$  and  $X_3^{(n)}$  is  $A_3^{(n)}$  then  $O$  is  $B_1^{(n)}$

in which  $A_1^{(1)}$ ,  $A_2^{(1)}$ , ...,  $B_1^{(1)}$ , etc. are linguistic terms, such as "high" and "low", and the value of the output  $O$  for each rule is given by the minimum of the membership levels multiplied by a weight.

In practice, with the neuro-fuzzy network of Figure 2, each second-layer output  $oL2(n)$  is equal to the minimum one among the products of the  $M$  outputs  $oL1(m)$  of the first-layer neurons 6 and a respective second-layer weight  $W_{FA}(m, n)$ .

In Figure 2 the above operation is represented by  $N$  second-layer neurons 7 which implement the equation

$$oL2(n) = \min_n \{W_{FA}(m, n) \cdot oL1(m)\} \quad (5)$$

Finally, the third layer corresponds to a defuzzification operation and supplies at output a discrete-type reconstruction weight  $oL3$ , using  $N$  third-layer weights  $W_{DF}(n)$ . The defuzzification method is that of the center of gravity (centroid) and is represented in Figure 2 by a third-layer neuron 8 supplying the reconstruction weight  $oL3$  according to the equation

$$oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)} \quad (6)$$

The reconstruction unit 4 then awaits a sufficient number of samples  $e(i)$  and corresponding reconstruction weights  $oL3(i)$  (at least  $2N + 1$ , corresponding to the width of a work window) and calculates an output sample  $u(i)$  as the weighted sum of the input samples  $e(i - j)$ , with  $j = 0 \dots 2N$ , using the reconstruction weights  $oL3(i - j)$ , according to the equation

$$u(i) = \frac{\sum_{j=0}^{2N} oL3(i - j) \cdot e(i - j)}{\sum_{j=0}^{2N} e(i - j)} \quad (7)$$

The training unit 5 operates only in an initial learning stage, when an input signal  $In$  having a known configuration is supplied to the filter 1, the output signal  $Out$  obtained is compared with a target signal  $Tg$ , and the distance between the obtained signal  $Out$  and the target signal  $Tg$  is evaluated on the basis of a fitness function. This fitness function may be, for example, the quadratic signal/noise ratio having the following expression:

$$SNR = \sum_{i=1}^T \frac{(Tg(i))^2}{(e(i) - Tg(i))^2} \quad (8)$$

in which  $T$  is the total number of input samples  $e(i)$ .

Using the fitness function and applying a method for function optimization, such as the gradient-descent method, a genetic algorithm, the simulated-annealing method, and random search, the first-layer weights (mean value  $W_m(l, k)$  and variance  $W_v(l, k)$  of the gaussian membership functions) and the third-layer weights  $W_{df}(n)$  are modified, and a new fitness evaluation is performed. The second-layer weights  $W_{FA}(m, n)$  are instead randomly initialized and are not modified. The learning process is iterated until a preset value of the fitness function is achieved or until a preset number of genetic algorithms have been generated or a preset number of steps of the selected optimization algorithm has been performed.

In this way, the neuro-fuzzy network 3 implements an adaptive algorithm and overcomes the limitations of neural networks or of fuzzy systems considered separately. In fact, fuzzy systems do not have learning capabilities and, if the selection of the fuzzy rules is not accurate, the fuzzy algorithm does not behave satisfactorily. Instead, using the neuro-fuzzy network 3 and carrying out a prior learning step it is possible to approach a signal having a complex mathematical structure, for example the voice in a noisy environment, without any prior knowledge of the mathematical laws governing the said system.

Operation of the filter 1 of Figure 1 is described in detail hereinafter with reference to Figures 3a and 3b.

Initially, the filter 1 is initialized, step 10, as is a window counter  $p$ , step 11. Then the window counter  $p$  is incremented by  $(2N + 1)$ , equal to the number of input samples  $e(i)$  in a work window, step 12, and the input samples of a work window ranging between  $e(p)$  and  $e(p + 2N)$  are loaded, step 13. A sum-of-samples counter  $h$  is

initialized at 0, and a sum-of-samples variable  $S(0)$  is initialized with the value of the input sample  $e(p)$ , step 14. Next, the sum-of-samples counter  $h$  is incremented by one unit, step 15, and the sum-of-samples variable  $S(h)$  is incremented with the value of a sample  $e(p + h)$ , step 20. After the sum of  $(2N + 1)$  input samples (output YES from step 21), the sum-of-samples variable  $S(h)$  is equal to the sum of all the input samples  $e(i)$  and may be directly used for calculating the signal features.

Subsequently, using equations (1), (2) and (3) and the value of the sum-of-samples variable  $S(h)$ , the signal features  $X_1$ ,  $X_2$ ,  $X_3$  are calculated for each sample  $e(p) - e(p + 2N)$  belonging to the considered work window, step 22. A sample counter  $i$  for the input samples within the considered work window is then reset to zero, step 23, and subsequently incremented by one unit, step 24; a feature counter  $l$  for the number of signal features used (in the present case, 3) is reset to zero, step 25, and subsequently incremented, step 26; and a function counter  $k$  for the number of membership functions used for each signal feature (in the present case, 2) is reset to zero, step 30, and subsequently incremented, step 31. Next, the first-layer weights (the mean value  $W_m(l, k)$  and the variance  $W_v(l, k)$ ) are loaded, step 32, and the first-layer outputs  $oL_1(l, k)$  are calculated in accordance with equation (4), step 33.

Loading of the first-layer weights  $W_m(l, k)$ ,  $W_v(l, k)$  and calculation of the first-layer outputs  $oL_1(l, k)$  are repeated for all the membership functions (output YES from step 34) and for all the signal features (output YES from step 35).

Next, a second-layer neuron counter  $n$  is reset to zero, step 39, and incremented by one unit, step 40; an algorithm counter  $m$  is reset to zero, step 41, and incremented, step

42; and the second-layer weights  $W_{FA}(m, n)$  for the  $n$ -th second-layer neuron 7 are loaded, steps 43, 44. Then, the second-layer output  $oL2(n)$  for the  $n$ -th second-layer neuron 7 is calculated using equation (5), step 45, and the procedure of steps 40-45 is repeated for all the  $n$  second-layer neurons 7.

At the end (output YES from step 50), the second-layer neuron counter  $n$  is reset again, step 51, and incremented by one unit, step 52; all the third-layer weights  $W_{DF}(n)$  are loaded in succession, steps 53, 54; at the end of loading (output YES from step 54) the reconstruction weight  $oL3(i)$  is calculated in accordance with equation (6), step 55.

The reconstruction weights  $oL3(i)$  thus calculated and the respective input samples  $e(i)$  are stored, step 60, and the procedure described for steps 24-60 is repeated for successive input samples  $e(i)$  until  $(2N + 1)$  input samples  $e(i)$  are reached (output YES from step 61).

Next, a sum counter  $j$  is reset to zero, step 62, and the input sample  $e(i)$  is added to the previous  $2N$  input samples  $e(i - j)$ , and the sum is stored in a variable  $E$ , step 63.

After the sum of  $(2N + 1)$  input samples has been carried out, an output sample  $u(i)$  is calculated in accordance with equation 7, step 65.

The entire cycle described by steps 12-65 is repeated as long as input samples  $e(i)$  are present; at the end (output YES from step 70), the sample processing procedure terminates.

According to another aspect of the invention, filtering is based on a multi-resolution analysis obtained through a filter bank in phase quadrature. Wavelet theory furnishes

the theoretical basis for multi-resolution.

As is known, a multi-resolution analysis defines a set of nested subspaces of a square summable function space, i.e., the space of the finite-energy functions, widely known in physics and electrical engineering. On the basis of the above analysis, a projection of a function in one of these subspaces isolates the "roughest" details of the function, whilst projection of the function in the orthonormal complement of this subspace isolates the "finer" details of the function. The procedure may be iterated to obtain a pyramid. From wavelet theory it is known that the operation may be performed using a bank of FIR filters, in which each FIR filter is followed by a subsampler). The signal thus split into sub-bands can be reconstructed using a bank of complementary filters, each of which is provided with a sample incrementer (upsampler).

A first embodiment of the above-mentioned solution is shown in Figure 4 and regards a multi-resolution filter generating two subspaces and using a first pair of Finite Impulse Response Quadrature Mirror Filters (FIR QMFs) for the analysis and a second pair of FIR QMFs for the synthesis.

In detail, the multi-resolution filter 80 receives, on an input 81, an input signal  $s_i(t)$ . The input signal  $s_i(t)$  is supplied to two input FIR filters  $H_0$ ,  $H_1$  which perform a convolution operation so to output a corresponding filtered signal  $\tilde{e}_0$ ,  $\tilde{e}_1$  equal to:

$$\tilde{e}_0(t) = \sum_{r=0}^M h_0(r) \cdot i(t-r) \quad (9)$$

$$\tilde{e}_1(t) = \sum_{r=0}^M h_1(r) \cdot i(t-r) \quad (10)$$

in which  $M$  is the order of the filters  $H_0$ ,  $H_1$ ;  $\tilde{e}_0(t)$ ,  $\tilde{e}_1(t)$  is the  $t$ -th sample of the respective output sequence;  $s_i(t)$  is the  $t$ -th sample of the input sequence;  $h_0(r)$ ,  $h_1(r)$  is the

$t$ -th tap of the input FIR filter  $H_0, H_1$ , in which  

$$h_1(r) = (-1)^{r-1} h_0(2I - r + 1)$$

where  $I$  is an integer.

5 The input FIR filters  $H_0, H_1$  have transfer functions shown in  
 Figure 5 and thus define, respectively, a low-pass filter  
 and a high-pass filter, and have the structure shown in  
 Figure 6, where the generic tap  $h(r)$  corresponds to  $h_0(r)$  or  
 to  $h_1(r)$ , according to whether it is the input FIR filter  $H_0$   
 10 or the input FIR filter  $H_1$ .

The outputs of the input FIR filters  $H_0, H_1$  are each  
 connected to a respective subsampling unit 84, 85 which  
 discards the odd samples from the input signal  $\tilde{z}_0(t), \tilde{z}_1(t)$   
 15 and retains only the even samples, generating a respective  
 signal  $e_0(t), e_1(t)$ . The outputs of the subsampling units 84,  
 85 are each connected to a respective neuro-fuzzy filter 86,  
 87. Both of the neuro-fuzzy filters 86, 87 have the  
 structure shown in Figure 1.

20 The output signals  $u_0(t), u_1(t)$  of the neuro-fuzzy filters  
 86, 87 are each supplied to a respective upsampler 88, 89  
 which generates a respective output signal  $u_{0u}(t), u_{1u}(t)$  by  
 entering a zero sample between each pair of samples of the  
 25 respective output signal  $u_0(t), u_1(t)$  of the neuro-fuzzy  
 filters 86, 87. The outputs of the of the upsampling units  
 88, 89 are each connected to a respective output FIR filter  
 $G_0, G_1$ . These filters too have each a respective transfer  
 function given by equation (9) and equation  
 30 (10), respectively.

Finally, the output signals of the output FIR filters  $G_0, G_1$   
 are added together, sample by sample, by an adder 92.

35 Figure 7 shows a multi-resolution filter 95 using eight  
 subspaces. In detail, the input signal  $s_1(t)$  is initially  
 supplied to two first synthesis FIR filters  $H_{001}, H_{102}$ ,

respectively of the low-pass type and of the high-pass type, and is then subsampled in two first subsampling units 96, 97, in a similar way as described for the units 85, 86 of Figure 4. The sequences of samples thus obtained are each  
 5 supplied to two synthesis filters (and hence altogether to four second synthesis FIR filters  $H_{011}$ ,  $H_{112}$ ,  $H_{013}$ , and  $H_{114}$ ). The outputs of the second synthesis FIR filters  $H_{011}$ ,  $H_{112}$ ,  $H_{013}$ , and  $H_{114}$  are then supplied to four second subsampling units 100-103, and each sequence thus obtained is supplied  
 10 to two third synthesis FIR filters (and hence altogether to eight third synthesis FIR filters  $H_{021}$ ,  $H_{122}$ ,  $H_{023}$ ,  $H_{124}$ ,  $H_{025}$ ,  $H_{126}$ ,  $H_{027}$ ,  $H_{128}$ ), generating eight sequences of samples. The eight sample sequences are then supplied to eight third subsampling units 107-114 and processed in respective neuro-  
 15 fuzzy filters 120-127 having the structure illustrated in Figure 1. The sample sequences present on the outputs of the neuro-fuzzy filters 120-127 are then incremented via upsampling units 130-137 and supplied to respective first synthesis FIR filters  $G_{021}$ ,  $G_{122}$ ,  $G_{023}$ ,  $G_{124}$ ,  $G_{025}$ ,  $G_{126}$ ,  $G_{027}$ , and  
 20  $G_{128}$ . The sample sequences thus obtained are added up two by two through four adders 140-143 (with a reverse process with respect to the one followed downstream of the second subsampling units 100-103), supplied to four upsampling units 146-149, and filtered again through four second  
 25 synthesis FIR filters  $G_{011}$ ,  $G_{112}$ ,  $G_{013}$ , and  $G_{114}$ .

The sample sequences thus obtained are added up two by two through two adders 150, 151 (according to a reverse process with respect to the one followed downstream of the first  
 30 subsampling units 96, 97), incremented by two upsampling units 154, 155, filtered through two third synthesis FIR filters  $G_{001}$  and  $G_{102}$ , and finally summed in an adder 160 so as to supply the output signal  $s_o(t)$ .

35 Figure 8 shows a flowchart of the sequence of steps performed using a multi-resolution filter with a preset



arbitrary number of splittings into subspaces.

Initially, the samples of the input signal  $s_i(t)$  are loaded, step 160; then a space split counter  $j$  is initialized at  $-1$ , step 161, and incremented by one unit, step 162; a subspace counter  $k$  is initialized at zero, step 163, and incremented by one unit, step 164. Then the samples of the input signal  $s_i(t)$  are filtered using the filter  $H_{0jk}$  (thus, at the first iteration, using the filter  $H_{001}$ ), step 165, and the filtered samples are downsampled, step 166. Next, the subspace counter  $k$  is incremented, step 170; the samples of the input signal  $s_i(t)$  are filtered using the filter  $H_{1jk}$  (thus, at the first iteration, using the filter  $H_{102}$ ), step 171, and the filtered samples are downsampled, step 172.

It is checked whether the subspace counter  $k$  is equal to  $2^{(j+1)}$ ; if it is not, the cycle comprising steps 164-172 is repeated (in the iterations following on the first, filtering is performed on the samples obtained in the previous iteration with the filter  $H_{0(j-1)(k-2)}$  or  $H_{1(j-1)(k-2)}$ ); if it is (output YES from step 173), it is checked whether the splitting into subspaces is terminated (i.e., whether the space split counter  $j$  has reached the preset maximum), step 174. If it has not, the procedure returns to step 162; if it has, all the obtained sample sequences are filtered using the neuro-fuzzy filter of Figure 1, step 175.

Next, the space split counter  $j$  is initialized at its maximum value plus one unit, step 180, and then decreased by one unit, step 181, the subspace counter  $k$  is initialized at zero, step 182, and incremented by one unit, step 183. Next, the first sample sequence at output of the first neuro-fuzzy filter (120, in Figure 7) is upsampled, step 184, and filtered using a filter  $G_{0jk}$  (thus, at the first iteration, using the filter  $G_{021}$ ), step 185. Then, the subspace counter  $k$  is incremented again, step 190; a second sample sequence

at the output of a second neuro-fuzzy filter (121, in Figure 7) is upsampled, step 191, and filtered using a filter  $G_{1jk}$  (thus, at the first iteration, using the filter  $G_{121}$ ), step 192. The samples at the output of the filters  $G_{0jk}$  and  $G_{1jk}$  are then summed, step 193.

It is then checked whether the subspace counter  $k$  is equal to  $2^{(j+1)}$ , step 194; if it is not, the cycle comprising steps 183-193 is repeated (processing the sample sequences at the output of the subsequent neuro-fuzzy filters); if it is (output YES from step 194), it is checked whether the end has been reached, step 195; if it has not, the procedure returns to step 181, decreasing the space split counter  $j$  and processing the sample sequences previously upsampled, filtered and summed. The loop defined by steps 181-194 is repeated until a single sequence of samples is obtained, corresponding to the output signal  $s_o(t)$ , output YES from step 195.

Figure 9 shows the hardware implementation of the neuro-fuzzy filter 1 of Figure 1. In detail, the neuro-fuzzy filter 1 comprises a data memory 200, three work memories 201-203, a signal features calculation module 205, a control unit 206, a first-layer output memory unit 207, a second-layer output calculating unit 208, a reconstruction-weight calculating unit 209, and a reconstruction unit 210.

The data memory 200 stores the  $(2N + 1)$  samples  $e(i)$  of each work window and comprises  $(2N + 1)$  registers, each having 16 bits.

The work memories 201-203 are nonvolatile memories, for example ROM, PROM, EPROM, EEPROM or flash memories.

In particular, the first work memory 201 stores the first signal feature  $X_1(i)$  and comprises  $(2N + 1)$  sixteen-bit

memory locations. Since the value of the first signal feature  $X1(i)$  for the  $i$ -th sample is constant in the various sample windows, as is evident from equation (1), the contents of the first work memory 201 must not be modified during the learning step or during operation of the neuro-fuzzy filter 1, and the first work memory 201 may be implemented using any one of the technologies referred to above.

The second work memory 202 stores the values of the two gaussian curves described by equation (2) according to the values of the signal features  $X1$ ,  $X2$ , and  $X3$ . Since the values of these gaussian curves depend upon the second-layer weights  $W_m(1, k)$ ,  $W_v(1, k)$ , when a learning step is provided, the second work memory 202 must be of the programmable type, for example of the EPROM, EEPROM or flash type. To avoid the use of a memory of excessive size, the gaussian functions (which represent the membership functions of the signal features  $X1$ ,  $X2$ ,  $X3$ , as discussed above) are stored as discrete values, according to the desired level of accuracy. For example, if the membership functions have 256 values, with an accuracy of 16 bits per value, considering two sets of fuzzy rules for each signal feature  $X1$ ,  $X2$ , and  $X3$ , the second work memory 202 must have a storage capacity of  $256 \times 16 \times 6$  bits. The second work memory 202 is then addressed starting from the current values (corresponding to the  $i$ -th sample) of the signal features  $X1$ ,  $X2$ ,  $X3$  supplied by the first work memory 201 and by the signal features calculation module 205, and outputs (to the first-layer output memory unit 207) the values of the six first-layer outputs  $oL1(m)$ .

The third work memory 203 stores the second-layer weights  $W_{FA}(m, n)$  and the third-layer weights  $W_{DF}(n)$ . Since the third-layer weights  $W_{DF}(n)$  are generally modified during the learning step, the third work memory 203 is of a

programmable type, as is the second work memory 202. In detail, if there are  $M \times N$  second-layer weights  $W_{FA}(m, n)$  and  $N$  third-layer weights  $W_{DF}(n)$ , the work memory 203 comprises  $(16 \times M \times N + 16 \times N)$  bits.

5

The signal features calculation module 205 comprises a hardware network (not shown) represented as divided into a first part 205a for calculating the second signal feature  $X2(i)$  and a second part 205b for calculating the third signal feature  $X2(i)$ . In the signal features calculation module 205 the operations represented by equations (2) and (3) are performed, and this module comprises a sample memory (having  $2N + 1$  locations), a unit for calculating the average value  $av$ , a unit for calculating the maximum of the differences between all the input samples and the central sample  $\max(diff)$ , a unit for calculating the maximum of the differences between all the input samples and the average value  $\max(diff\_av)$ , and a unit for calculating the fractions defined by equations (2) and (3).

20

The first-layer output memory unit 207 comprises six registers 212 which store the first-layer outputs  $oL1(m)$  supplied by the second work memory 202.

25

The second-layer output calculating unit 208 comprises two modules, namely, a first multiplication module 208a and a minimum module 208b. In detail, the first multiplication module 208a includes six first multiplication units 213 each of which multiplies a respective first-layer output  $oL1(m)$  (supplied by the first-layer output memory unit 207) by  $n$  respective second-layer weights  $W_{FA}(m, n)$  (supplied by the second work memory 203); the second multiplication module 208b includes six minimum units 214 which, starting from the six respective products  $oL1(m) \times W_{FA}(m, n)$ , calculate the minimum thereof, supplying at the output a respective second-layer output  $oL2(n)$ .

35

The reconstruction-weight calculating unit 209 comprises two modules, namely, a second multiplication module 209a and a defuzzification module 209b. In detail, the second multiplication module 209a includes six second multiplication units 215 which multiply a respective second-layer output  $oL2(n)$ , supplied by the second-layer output calculating unit 208, by a respective third-layer weight  $W_{DF}(n)$ , supplied by the third work memory 203. The defuzzification module 209b calculates the reconstruction weights  $oL3$ , adding the products supplied by the second multiplication module 209a, adding together the second-layer outputs  $oL2(n)$ , and calculating the ratio between the two sums in accordance with equation (6).

The reconstruction unit 210 stores the reconstruction weights  $oL3(i)$  as these are supplied by the reconstruction-weight calculating unit 209 and, as soon as it has stored  $2N + 1$  reconstruction weights  $oL3$ , it calculates the output sample  $u(2N + 1)$  in accordance with equation (7), also using the values of the  $2N + 1$  samples supplied by the data memory 200. Subsequently, upon receipt of the next reconstruction weight  $oL3(2N + 2)$ , it calculates the output sample  $u(2N + 2)$  using also the previous  $2N$  reconstruction weights  $oL3$  and as many samples coming from the data memory 200, in accordance with equation (7).

The control unit 206 determines the processing sequence and data loading/transfer between the various units and modules. To calculate a single output sample, the control unit repeats the sequence of steps of the fuzzy algorithm  $2N + 1$  times, updates the data memory 200 at the end of  $2N + 1$  cycles, and controls loading of successive  $2N + 1$  samples.

The advantages of the method and filter illustrated herein are described below. First, the method and filter reduce the

- 20-

noise of the input signal, whether the noise in question is of a white type or of a colored type, and enable separation of signals having different features. The filter preserves the steep edges of the signals without causing any losses of signal features, as is evident from a comparison between Figures 10a, 10b, and 10c. In particular, Figure 10a is a plot of a non-noisy voice signal (namely, a signal fragment corresponding to the vowel "e", with sampling at 44.1 kHz and a 16-bit A-D conversion resolution); Figure 10b is a plot the same signal as in Figure 10a, in presence of white noise; and Figure 10c shows the result of filtration of the signal of Figure 10b using the filter device 95 of Figure 7.

Since the filter can be trained, it can be adapted to a specific type of initial signal and can be subsequently modified if so required. For example, the filter can be initially adapted to a first type of acoustic signal (for instance, a male voice with white noise, training being performed with two signals, one in the presence of white noise and one in the absence of noise). Then, the learning step can be repeated for signals of a different type, for example a female voice. The new weights thus calculated can replace the previous ones or be stored in addition to the previous ones; in the latter case, the control unit 206 controlling the filter 1 can control sending, to the units 207-209 (Figure 9), either of the first or of the second samples according to the use of the filter.

Finally, it is clear that numerous variations and modifications may be made to the method and filter described and illustrated herein, all falling within the scope of the invention, as defined in the annexed claims.

22. 01. 2001

(40)

## CLAIMS

1. A filtering device (80; 95) for filtering signals having steep edges, characterized by at least one neuro-fuzzy filter (1).

5

2. A filtering device according to Claim 1, characterized in that said neuro-fuzzy filter (1) comprises:

a signal-feature calculating unit (2) receiving input samples ( $e(i)$ ) of a signal ( $In$ ) to be filtered and generating signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ );

10

a neuro-fuzzy network (3) receiving said signal features and generating reconstruction weights ( $oL3(i)$ ); and

a moving-average reconstruction unit (4) receiving said input samples ( $e(i)$ ) and said reconstruction weights ( $oL3(i)$ ), and generating output samples ( $u(i)$ ) from said input samples and said reconstruction weights.

15

3. A filtering device according to Claim 2, characterized in that said signal-feature calculating unit (2) comprises a memory (200) storing a first plurality of input samples ( $e(i)$ ) forming a first window, and a feature-calculating network (201, 205) receiving said first plurality of input samples and supplying a plurality of features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) for each one of said input samples.

20

25

4. A device according to Claim 3, characterized in that said feature-calculating network (201, 205) comprises first feature-providing means (201) which generate, for each of said input samples ( $e(i)$ ), a first signal feature ( $X1(i)$ ) correlated to a position of said input sample in said first window; second feature-providing means (205a) which generate, for each of said input samples ( $e(i)$ ), a second signal feature ( $X2(i)$ ) correlated to the difference between said input sample and a central sample ( $e(N)$ ) in said first window; and third feature-providing means (205b) which generate, for each of said input samples ( $e(i)$ ), a third

30

35

signal feature ( $X3(i)$ ) correlated to the difference between said input sample and an average sample value in said first window.

- 5 5. A device according to Claim 4, characterized in that said first feature-providing means (201) generate said first signal feature ( $X1(i)$ ) for an input sample ( $e(i)$ ) according to the relation

$$X1(i) = \frac{|i - N|}{N}$$

- 10 wherein  $i$  is the position of said input sample ( $e(i)$ ) in said first window, and  $N$  is the position of a central sample in said first window.

- 15 6. A device according to Claim 5, characterized in that said first feature-providing means comprise a first work memory (201) storing said first signal feature ( $X1(i)$ ) for each one of said input samples ( $e(i)$ ).

- 20 7. A device according to any of Claims 4-6, characterized in that said second feature-providing means (205a) generate said second signal feature ( $X2(i)$ ) for an input sample ( $e(i)$ ) according to the relation

$$X2(i) = \frac{|e(i) - e(N)|}{\max(\text{diff})}$$

- 25 wherein  $e(N)$  is a central sample in said first window, and  $\max(\text{diff})$  is the maximum of the differences between all the input samples in said first window and said central sample.

- 30 8. A device according to any of Claims 4-7, characterized in that said third feature-providing means (205b) generate said third signal feature ( $X3(i)$ ) for an input sample ( $e(i)$ ) according to the relation

$$X3(i) = \frac{|e(i) - av|}{\max(\text{diff\_av})}$$

wherein  $av$  is the average value of the input samples in said first window, and  $\max(\text{diff\_av})$  is the maximum of the



differences between all the input samples in said first window and said average value av.

9. A device according to any of Claims 2-8, characterized in that said neuro-fuzzy network (3) comprises:  
fuzzification neurons (6, 202) receiving said signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) of an input sample ( $e(i)$ ) and generating first-layer outputs ( $oL1$ ) defining a confidence level of said signal features with respect to preset membership functions;  
fuzzy neurons (7) of an AND type, receiving said first-layer outputs and generating second-layer outputs ( $oL2$ ) deriving from fuzzy rules; and  
a defuzzification neuron (8) receiving said second-layer outputs and generating a reconstruction weight ( $oL3$ ) for each of said input samples ( $e(i)$ ), using a center-of-gravity criterion.

10. A device according to Claim 9, characterized in that said membership functions are gaussian functions, and said first-layer outputs are calculated according to the equation

$$oL1(l,k) = \exp\left(-\left(\frac{Xl - W_m(l,k)}{W_v(l,k)}\right)^2\right)$$

wherein  $oL1(l, k)$  is a first-layer output,  $Xl$  is a signal feature,  $W_m(l, k)$  is the mean value, and  $W_v(l, k)$  is the variance of a gaussian function.

11. A filtering device according to Claim 9 or 10, characterized in that it comprises two membership functions for each one of said signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ).

12. A filtering device according to any of Claims 9-11, characterized in that said first-layer neurons (6) comprise a second work memory (202) storing values of said first-layer outputs ( $oL1(l, k)$ ) for each value of said signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ).

13. A filtering device according to any of Claims 9-12, characterized in that said fuzzy rules are of an AND type, and said second-layer outputs (oL2) use the norm of the  
5 minimum.

14. A filtering device according to Claim 13, characterized in that said second-layer outputs are calculated according to the equation  
10 
$$oL2(n) = \min_n \{W_{FA}(m, n) \cdot oL1(m)\}$$

wherein oL2(n) is a second-layer output;  $W_{FA}(m, n)$  is a second-layer weight, and oL1(l, k) is a first-layer output.

15. A filtering device according to Claim 14, characterized in that said second-layer neurons (7) comprise a plurality of multiplication units (213) and a plurality of minimum modules (214) cascade-connected together.

16. A filtering device according to any of Claims 9-15, characterized in that said reconstruction weights are  
20 calculated according to the equation

$$oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)}$$

wherein oL3(n) is a reconstruction weight;  $W_{DF}(n)$  are third-layer weights, and oL2(n) is a second-layer output.

25

17. A filtering device according to any of Claims 9-16, characterized in that said moving-average reconstruction unit (4) receives a second plurality of input samples (e(i)) forming a second window, and a corresponding plurality of  
30 reconstruction weights (oL3(i)), and calculates each of said output samples (u(i)) according to the equation

$$u(i) = \frac{\sum_{j=0}^{2N} oL3(i-j) \cdot e(i-j)}{\sum_{j=0}^{2N} e(i-j)}$$

wherein  $e(i-j)$  is an  $(i-j)$ -th input sample, and  $oL3(i-j)$  is a reconstruction weight associated to an  $(i-j)$ -th input sample.

5

18. A filtering device according to any of Claims 9-17, characterized by a training unit (5) having a first input connected to said moving-average reconstruction unit (4) and receiving said output samples ( $u(i)$ ), a second input receiving a desired output signal ( $Tg$ ), and an output connected to said neuro-fuzzy network (3) to supply optimized weighting values.

19. A filtering device according to any of Claims 1-18, characterized by a first splitting stage ( $H_1, H_0, 84, 85; H_{001}, H_{102}, 96, 97$ ) generating at least two streams of samples to be filtered; one said neuro-fuzzy filter ( $86, 87; 120-127$ ) for each stream of samples to be filtered, each of said neuro-fuzzy filters generating a respective stream of filtered samples; and a first recomposition stage ( $88, 89, 92, G_0, G_1; 154, 155, 160, G_{001}, G_{102}$ ) receiving said streams of filtered samples and generating a single stream of output samples ( $s_o$ ).

20. A filtering device according to Claim 19, characterized by a plurality of further splitting stages ( $H_{011}-H_{128}, 100-114$ ) cascade-connected together and to said first splitting stage ( $H_{001}, H_{102}, 96, 97$ ), and a plurality of further recomposition stages ( $130-151, G_{021}-G_{114}$ ) cascade-connected to each other between said neuro-fuzzy filters ( $120-127$ ) and said first recomposition stage ( $154, 155, 160, G_{001}, G_{102}$ ).

21. A filtering device according to Claim 19 or 20, characterized in that said splitting stages ( $H_1, H_0, 84, 85;$

$H_{001}-H_{128}$ , 96-114) each comprise a first and a second analysis filters ( $H_1$ ,  $H_0$ ,  $H_{001}-H_{128}$ ) in phase quadrature to each other and receiving a stream of samples to be split, said first and a second analysis filters generating a respected stream of split samples, and a first and a second downsampling unit (84, 85; 96-114), each of which receives a respective stream of split samples, and in that said recombination stages (88, 89, 92,  $G_0$ ,  $G_1$ ; 130-160,  $G_{021}-G_{102}$ ) each comprise a first and a second upsampling units (88, 89; 130-137, 146-149, 154, 155), each first and a second upsampling units receiving a respective stream of samples to be incremented and generating a respective stream of incremented samples; a first and a second synthesis filters ( $G_0$ ,  $G_1$ ;  $G_{021}-G_{102}$ ) in quadrature with to each other and complementary to said analysis filters, each of said first and a second synthesis filters receiving a respective stream of incremented samples and generating a respective stream of partial samples; and an adder node (92; 140-143, 150, 151, 160) receiving said streams of partial samples and generating a stream of added samples.

22. A filtering device according to Claim 21, characterized in that said analysis filters ( $H_1$ ,  $H_0$ ;  $H_{001}-H_{128}$ ) are quadrature mirror filters (QMFs), and said synthesis filters ( $G_0$ ,  $G_1$ ;  $G_{021}-G_{102}$ ) are QMFs complementary to said analysis filters.

23. A filtering device according to Claim 22, characterized in that said quadrature mirror filters ( $H_1$ ,  $H_0$ ;  $H_{001}-H_{128}$ ;  $G_0$ ,  $G_1$ ;  $G_{021}-G_{102}$ ) are convolutive filters.

24. A method for reducing noise in a filter having sharp edges, characterized by a neuro-fuzzy filtering step.

25. A method according to Claim 24, characterized in that it comprises the steps of:

calculating signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) from input samples ( $e(i)$ ) of a signal ( $In$ ) to be filtered;

calculating reconstruction weights ( $oL3$ ) from said signal features using a neuro-fuzzy network (3); and

5 reconstructing, from said input samples and said reconstruction weights and using a moving-average filter, an output signal ( $Out$ ) including a plurality of output samples ( $u(i)$ ).

10 26. A method according to Claim 25, characterized by the steps of:

storing a first plurality of input samples ( $e(i)$ ) forming a first window; and

15 calculating, from said first plurality of input samples, a plurality of signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) for each of said input samples.

20 27. A method according to Claim 26, characterized in that said step of calculating a plurality of signal features for each of said input samples comprises the steps of:

calculating a first signal feature ( $X1(i)$ ) correlated to a position of said input sample ( $e(i)$ ) in said first window;

25 calculating a second signal feature ( $X2(i)$ ) correlated to the difference between said input sample and a central sample in said first window; and

calculating a third signal feature ( $X3(i)$ ) correlated to the difference between said input sample and an average sample value  $av$  in said first window.

30 28. A method according to Claim 26 or 27, characterized in that said step of calculating reconstruction weights ( $oL3$ ) comprises the steps of:

performing a fuzzification operation and calculating first-layer outputs ( $oL1$ ) defining confidence levels of said  
35 signal features ( $X1(i)$ ,  $X2(i)$ ,  $X3(i)$ ) with respect to preset membership functions;

performing a fuzzy AND operation and generating second-layer outputs (oL2) deriving from fuzzy rules, starting from said first-layer outputs; and

performing a defuzzification operation on said second-layer outputs and generating a reconstruction weight (oL3) for each one of said input samples (e(i)), using a center-of-gravity criterion of the.

29. A method according to Claim 28, characterized in that said membership functions are gaussian functions, and said first-layer outputs are calculated according to the equation

$$oL1(l,k) = \exp \left( - \left( \frac{Xl - W_m(l,k)}{W_v(l,k)} \right)^2 \right)$$

wherein oL1(l, k) is a first-layer output, Xl is a signal feature,  $W_m(l, k)$  is the mean value, and  $W_v(l, k)$  is the variance of a gaussian function.

30. A method according to Claim 28 or 29, characterized in that said second-layer outputs are calculated according to the equation

$$oL2(n) = \min_n \{ W_{FA}(m,n) \cdot oL1(m) \}$$

wherein oL2(n) is a second-layer output;  $W_{FA}(m, n)$  is a second-layer weight, and oL1(l, k) is a first-layer output.

31. A method according to any of Claims 28-30, characterized in that said reconstruction weights are calculated according to the equation

$$oL3 = \frac{\sum_{n=1}^N W_{DF}(n) \cdot oL2(n)}{\sum_{n=1}^N oL2(n)}$$

wherein oL3 is a reconstruction weight;  $W_{DF}(n)$  are third-layer weights, and oL2(n) is a second-layer output.

32. A method according to any of Claims 26-31, characterized in that said step of reconstructing using a moving-average

filter comprises the steps of:

receiving a second plurality of input samples ( $e(i)$ ) forming a second window, and a respective plurality of reconstruction weights ( $oL3(i)$ ); and

- 5 calculating each of said output samples according to the equation

$$u(i) = \frac{\sum_{j=0}^{2N} oL3(i-j) \cdot e(i-j)}{\sum_{j=0}^{2N} e(i-j)}$$

wherein  $e(i-j)$  is an  $(i-j)$ -th input sample, and  $oL3(i-j)$  is a reconstruction weight associated to an  $(i-j)$ -th input sample.

10

33. A method according to any of Claims 24-32, characterized by a training step for training weights used in said neuro-fuzzy filtering step.

15

34. A method according to Claim 33, characterized in that said training step comprises the steps of:

generating an input signal having a known configuration;  
filtering said input signal having a known configuration to  
20 obtain a test output signal;

comparing said test output signal with a desired signal ( $Tg$ ) to obtain a distance between said test output signal and said desired signal;

calculating a fitness function from said distance; and

- 25 optimizing said weights in accordance with said fitness function.

35. A method according to any of Claims 26-34, characterized by a multiresolution analysis whereby the signal is splitted  
30 into sub-bands through orthonormal wavelets.

36. A method according to Claim 35, characterized by the steps of:

splitting a stream of input samples into at least two

streams of samples to be filtered;

filtering each stream of samples to be filtered using a respective neuro-fuzzy filter (86, 87; 120-127) to obtain at least two streams of filtered samples; and

5 recomposing said streams of filtered samples to generate a single stream of output samples ( $s_o$ ).

37. A method according to Claim 36, characterized in that, before performing said step of filtering, said step of  
10 splitting is repeated a preset number of times, and in that, after performing said step of filtering, said step of recomposing is repeated said preset number of times.

38. A method according to Claim 36 or 37, characterized in  
15 that said step of splitting comprises the steps of:  
feeding a stream of samples to be split to two analysis filters in phase quadrature to each other;  
generating two streams of filtered split samples; and  
downsampling said streams of filtered split samples,  
20 and in that said step of recomposing comprises the steps of:  
upsampling streams of samples to be incremented, generating streams of incremented samples;  
filtering said streams of incremented samples using two synthesis filters in phase quadrature to each other and  
25 complementary to said analysis filters, generating streams of partial samples; and  
adding pairs of streams of partial samples and generating a stream of added samples.



22 01. 2001

(40)

## ABSTRACT

The filtering device (80) comprises a neuro-fuzzy filter (1; 80) and implements a moving-average filtering technique in which the weights for final reconstruction of the signal (oL3(i)) are calculated in a neuro-fuzzy network (3) according to specific fuzzy rules. The fuzzy rules operate on three signal features (X1(i), X2(i), X3(i)) for each input sample (e(i)). The signal features are correlated to the position of the sample in the considered sample window, to the difference between a sample and the sample at the center of the window, and to the difference between a sample and the average of the samples in the window. The filter device for the analysis of a voice signal comprises a bank of neuro-fuzzy filters (86, 87). The signal is split into a number of sub-bands, according to wavelet theory, using a bank of analysis filters including a pair of FIR QMFs ( $H_0$ ,  $H_1$ ) and a pair of downsamplers (85, 86); each sub-band signal is filtered by a neuro-fuzzy filter (86, 87), and then the various sub-bands are reconstructed by a bank of synthesis filters including a pair of upsamplers (88, 89), a pair of FIR QMFs ( $G_0$ ,  $G_1$ ), and an adder node (92).

Figures 1 and 4

**THIS PAGE BLANK (USPTO)**

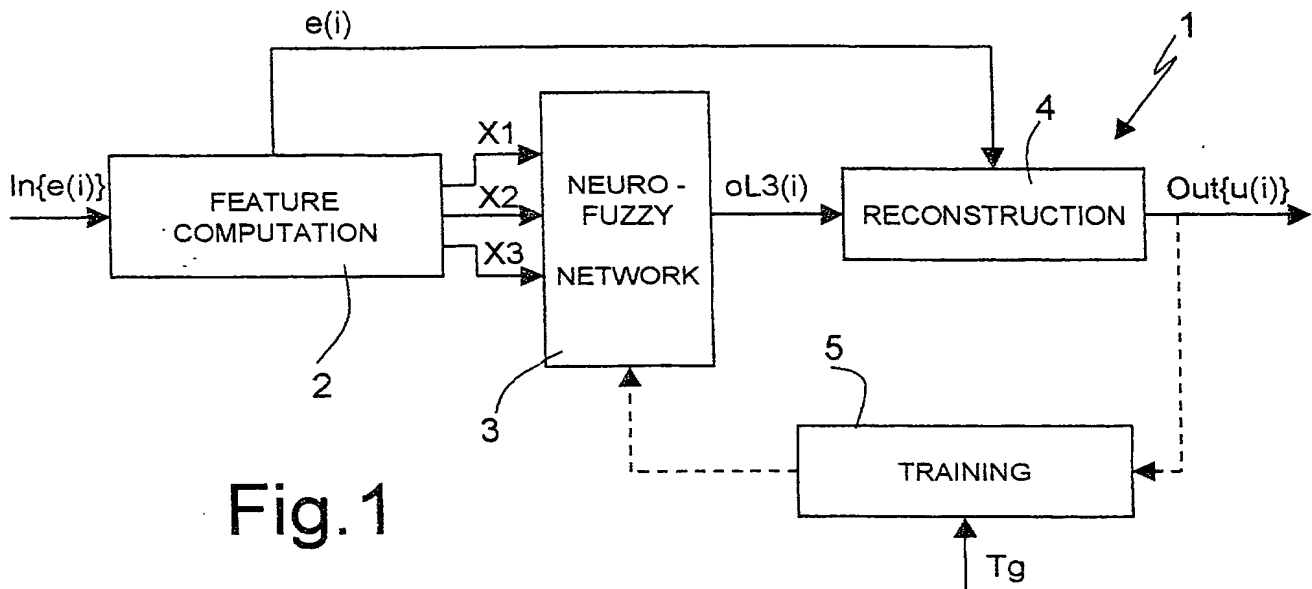


Fig.1

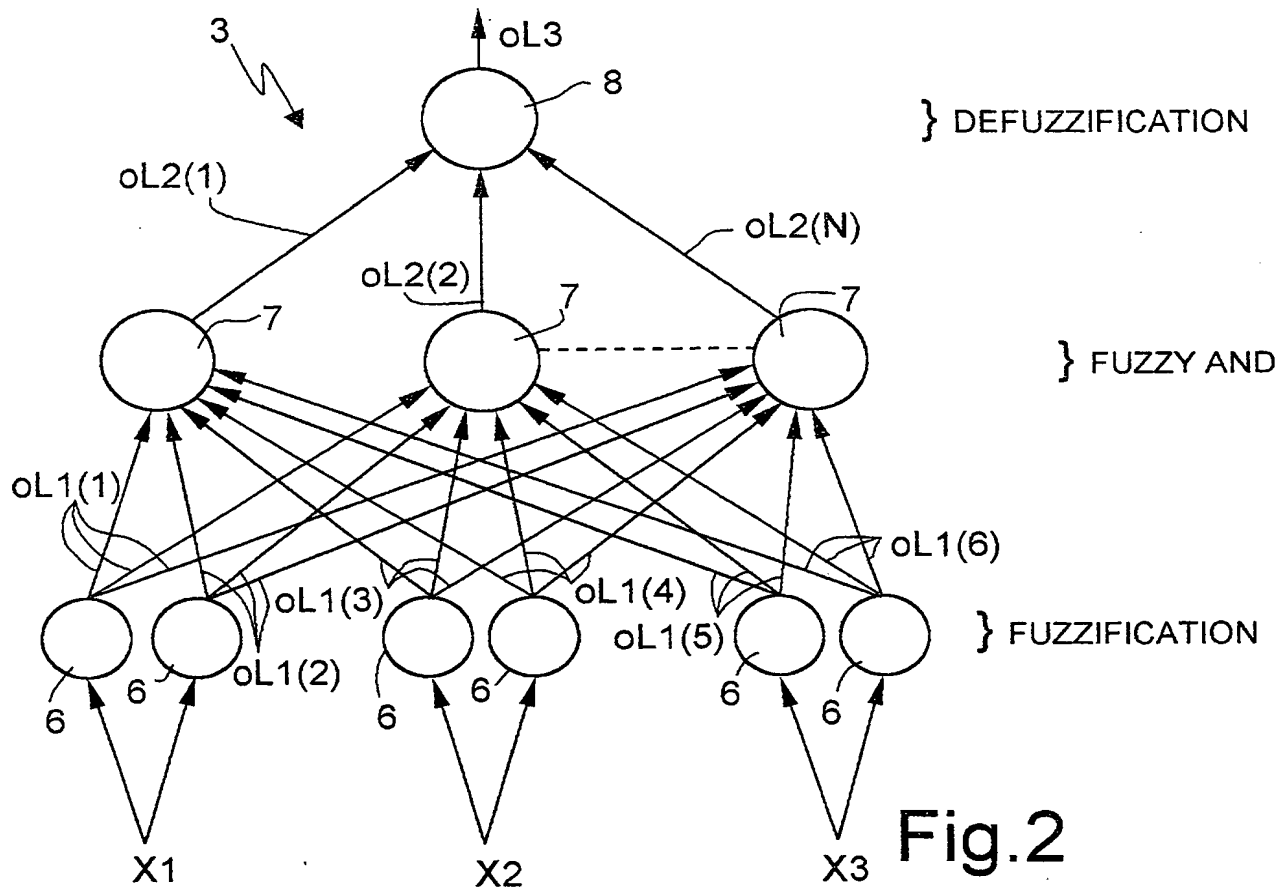


Fig.2

2 / 8

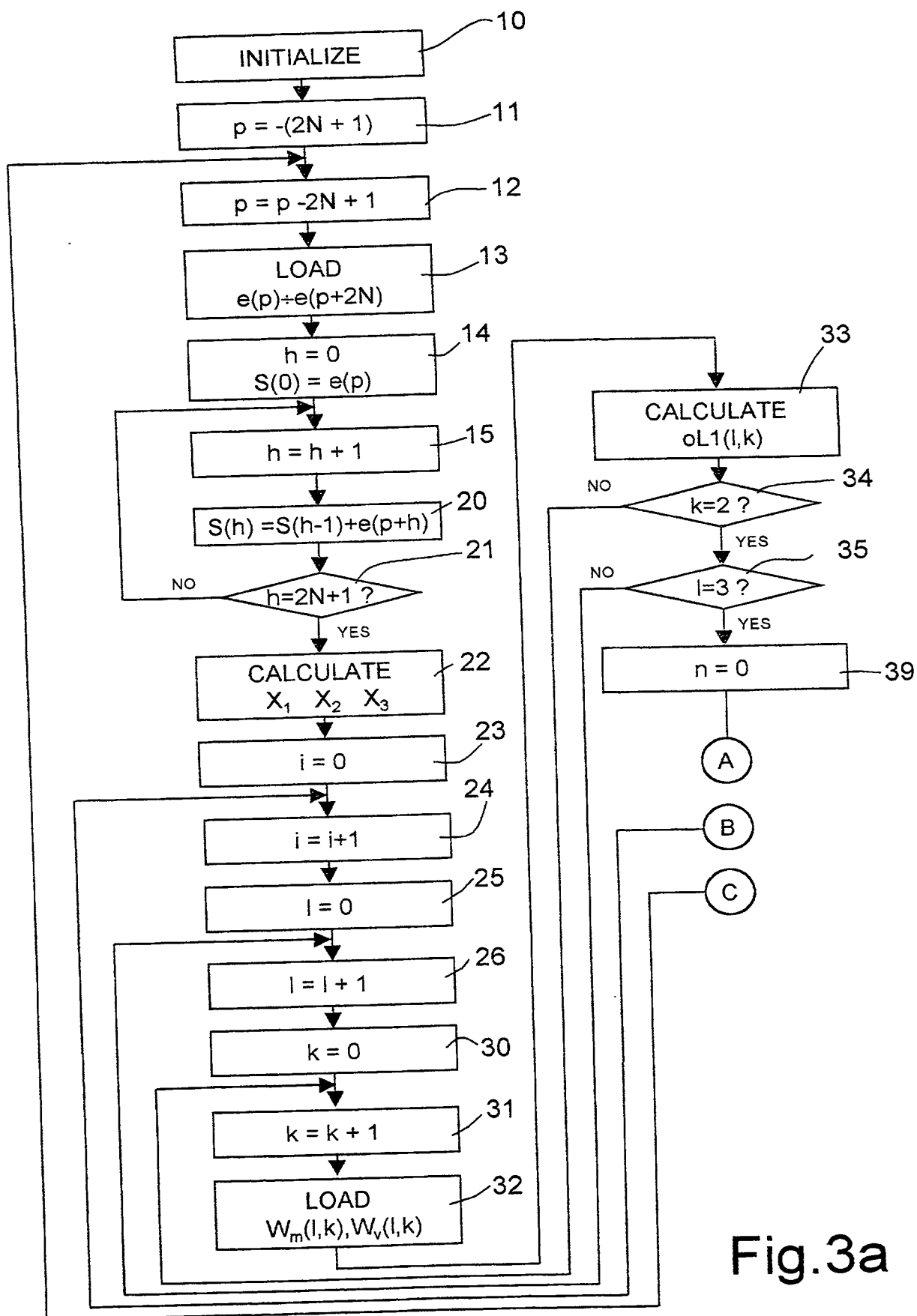


Fig.3a

3 / 8

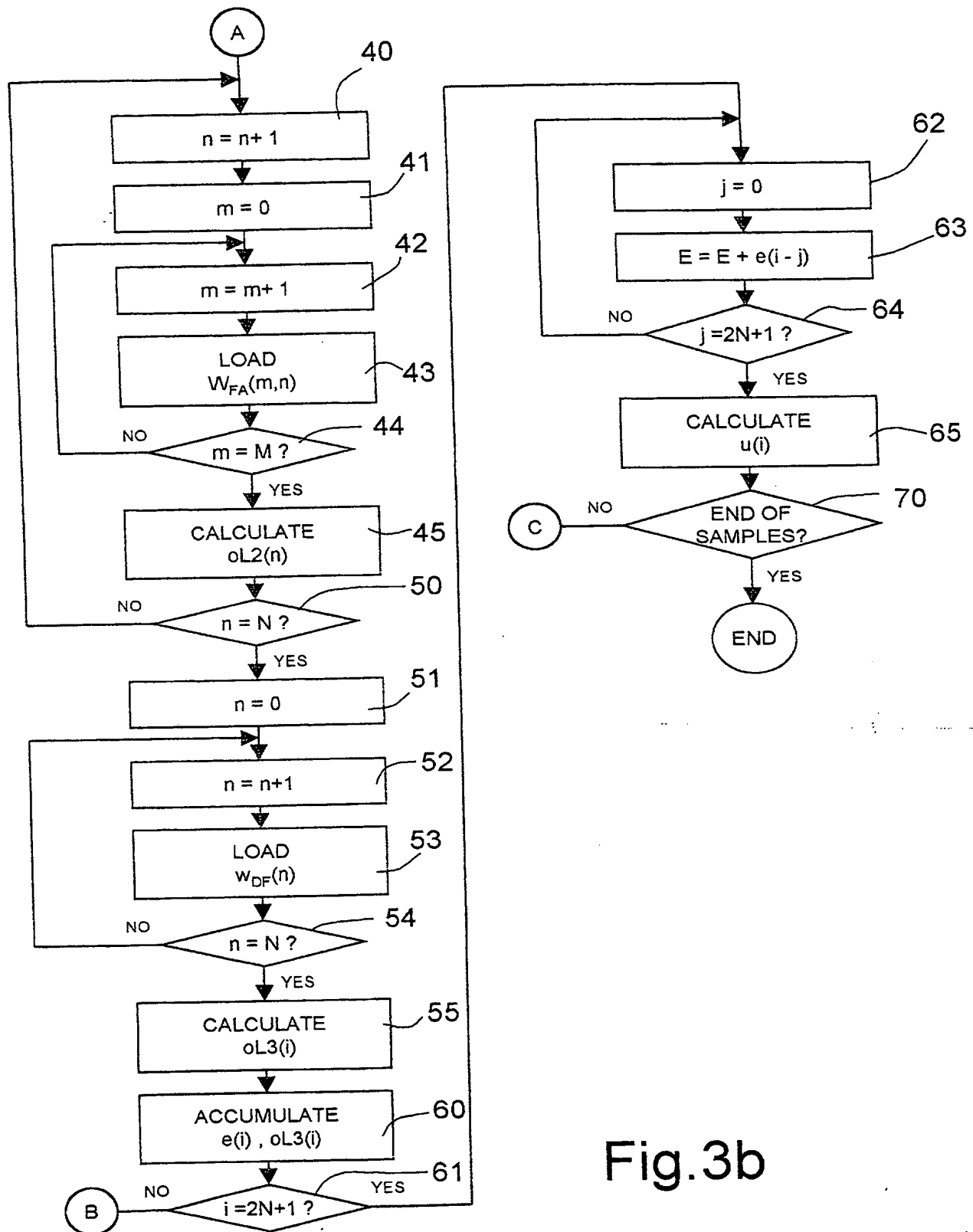


Fig.3b

4 / 8

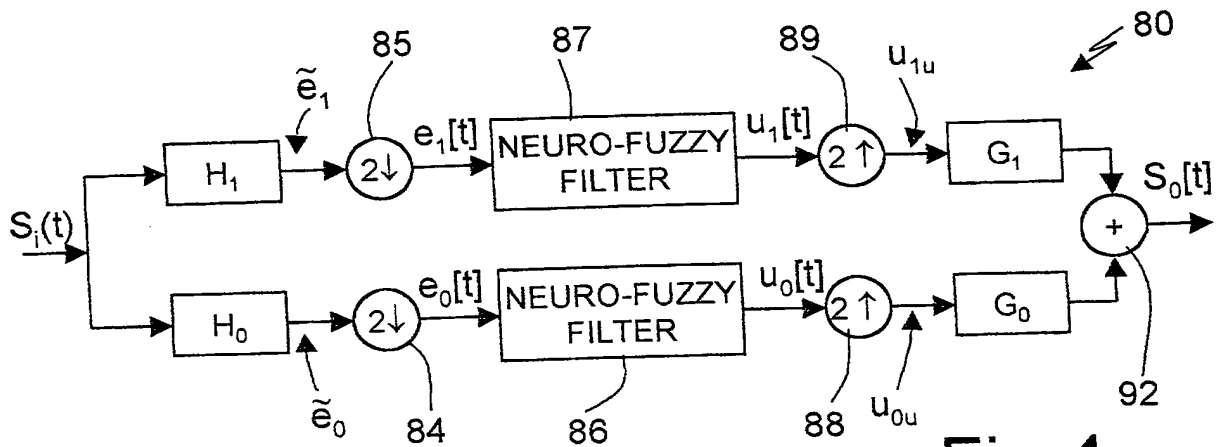


Fig.4

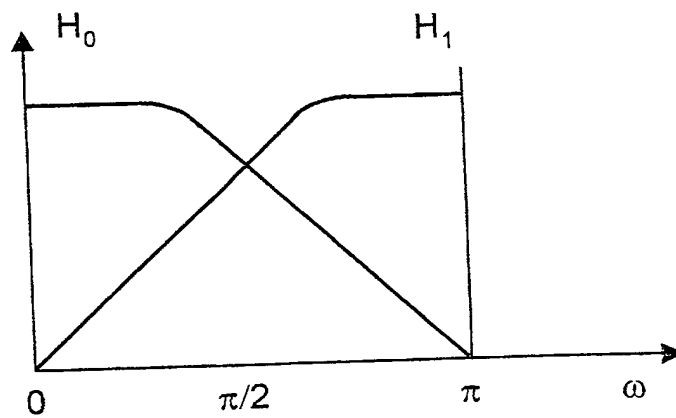


Fig.5

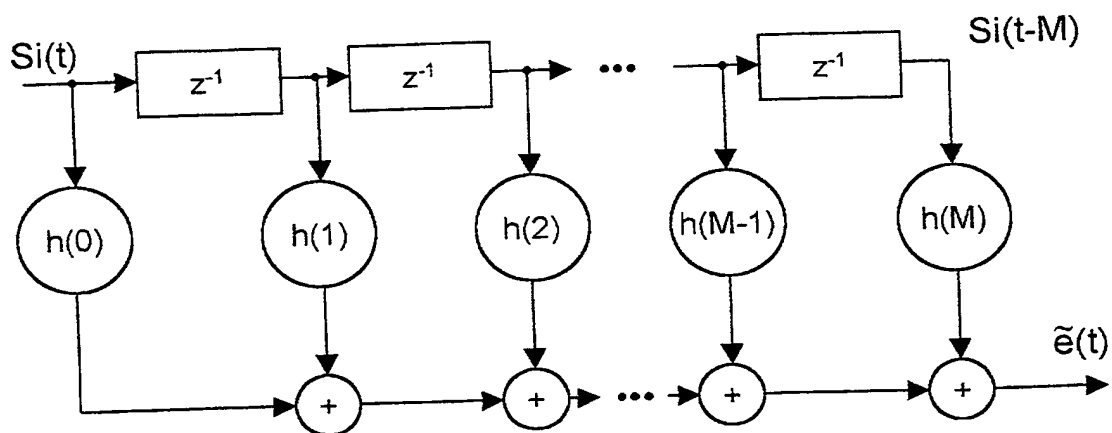


Fig.6

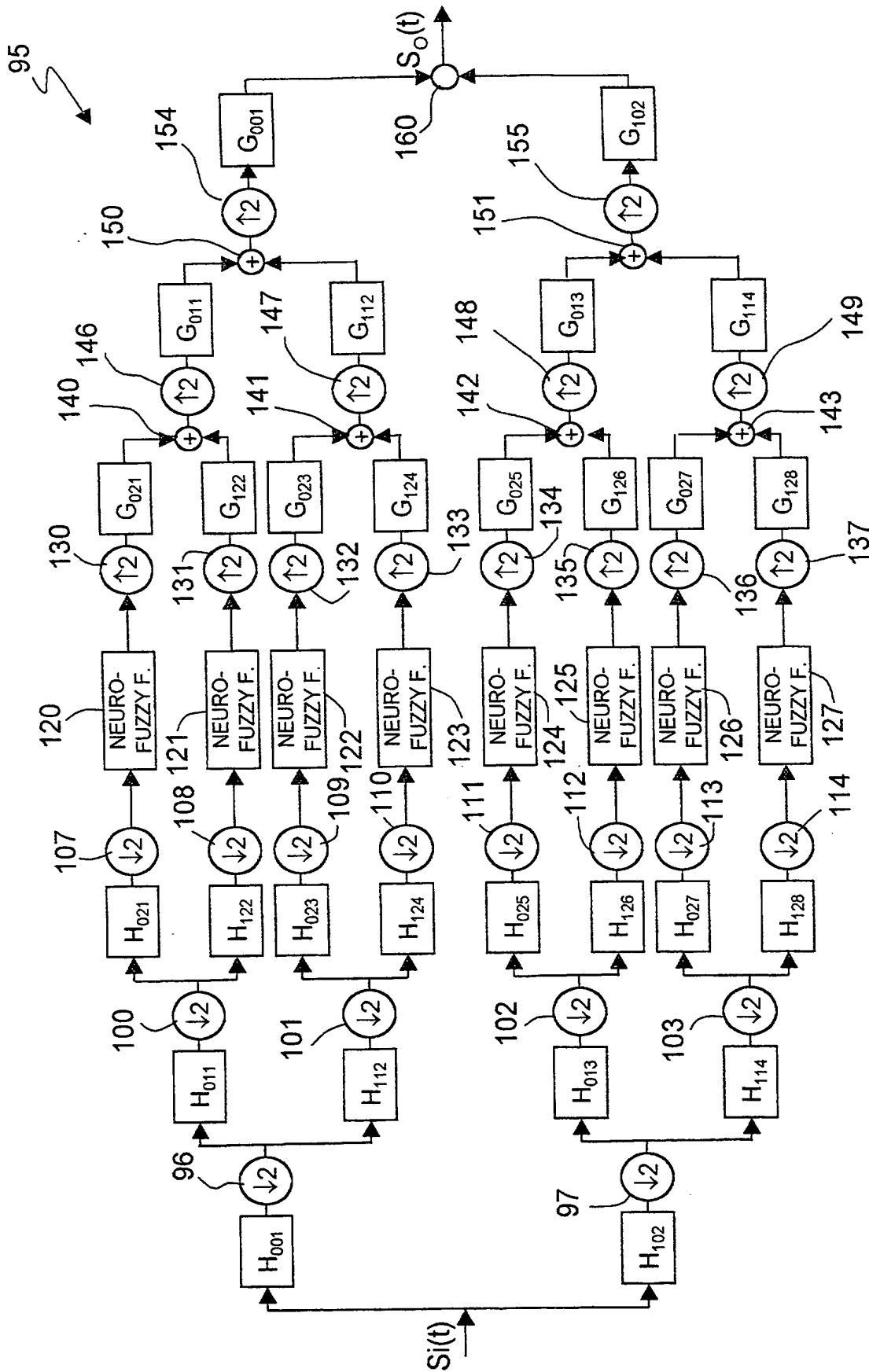


Fig.7

6 / 8

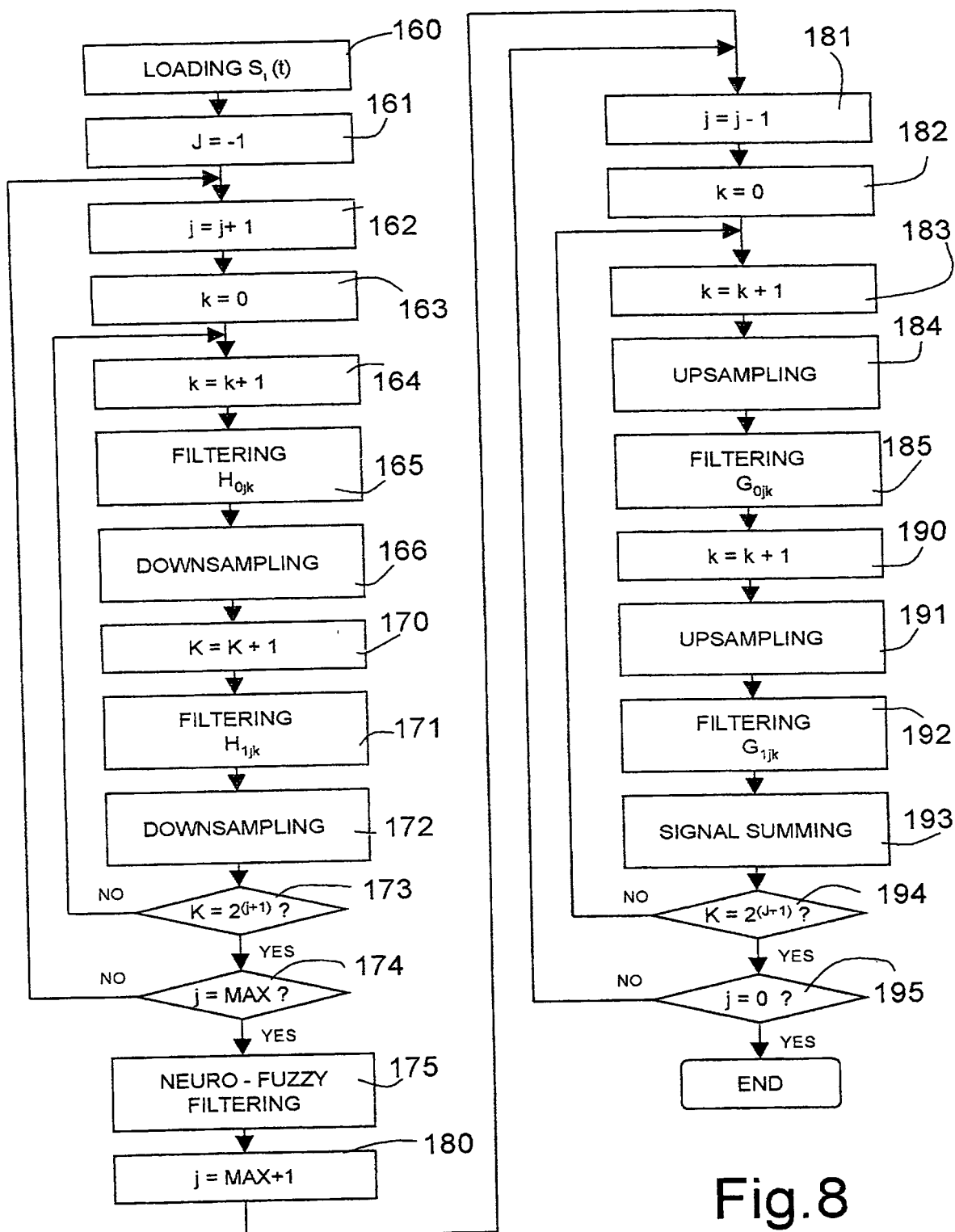


Fig.8



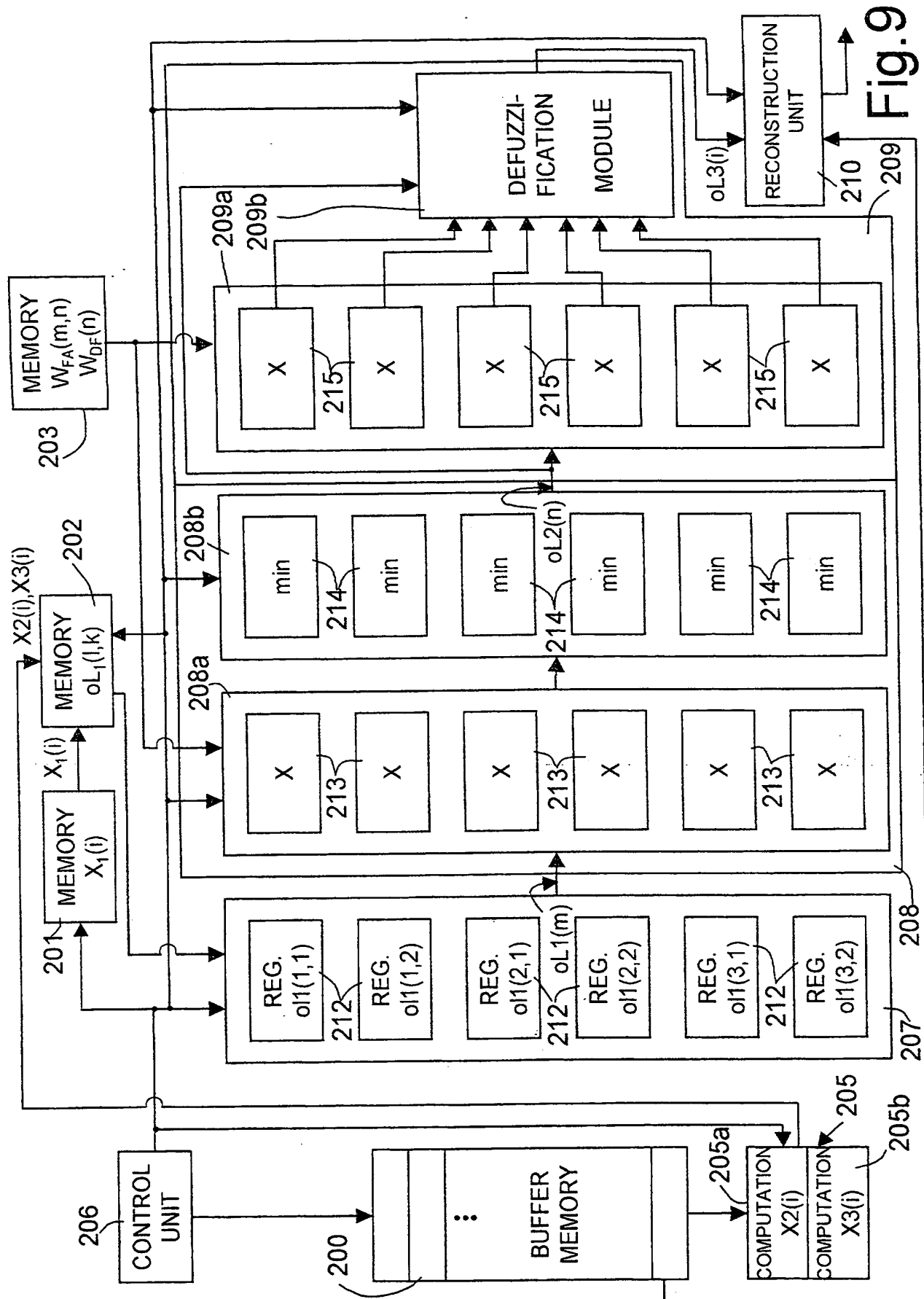


Fig.9

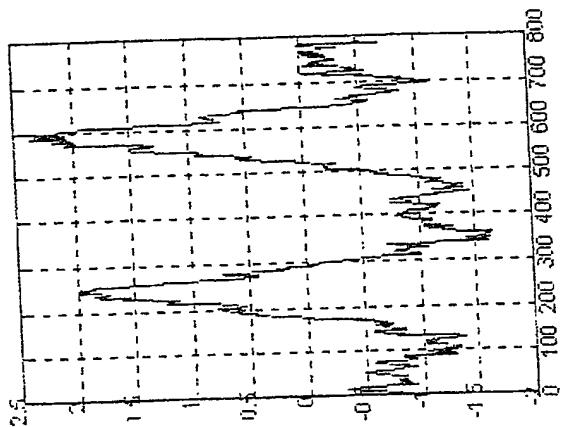


Fig. 10a

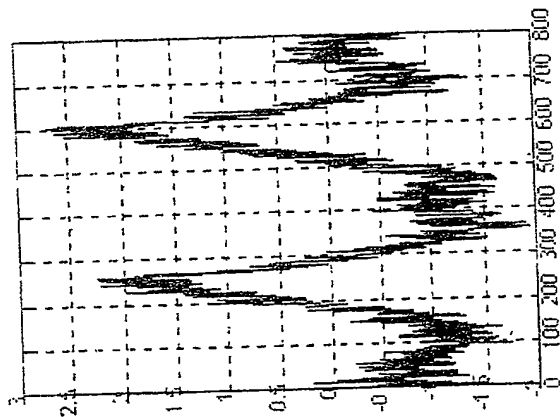


Fig. 10b

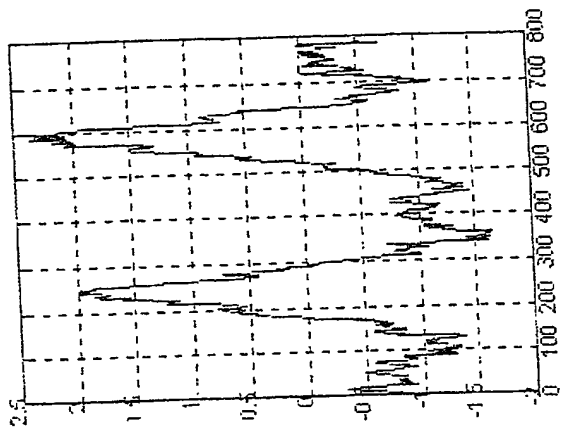


Fig. 10c

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINE(S) OR MARK(S) ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**